National Défense
Defence nationale

GROUND TERMINAL PROCESSOR INTERFACE BOARD FOR SKYNET UPLINK SYNCHRONIZATION TRIALS

by

Caroline Tom

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
REPORT NO. 1321

Canada

November 1997
Ottawa

National Défense
Defence nationale

GROUND TERMINAL PROCESSOR INTERFACE BOARD FOR SKYNET UPLINK SYNCHRONIZATION TRIALS

by

Caroline Tom
MilSat Communications Group
Space Systems & Technology Section

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
REPORT NO. 1321

PROJECT
5CA11

November 1997
Ottawa

Abstract

A ground terminal (GT) simulator subsystem is being developed at Defence Research Establishment Ottawa (DREO) as part of the in-house work examining the aspects of uplink synchronization for extremely-high-frequency (EHF) spread spectrum satellite communications (SATCOM). Requirements of the GT subsystem include the generation of hop clock and data clock signals, and the interface between the GT processor and a hopping synthesizer controller (HSC) for commanding the HSC and transmitting data. A GT processor interface (i/f) board was designed and fabricated at DREO to satisfy these requirements. This report describes the functions of the i/f board and specific requirements related to the uplink synchronization experiments and the interface to the HSC. The i/f board is a printed circuit board which is contained in a backplane chassis and is driven by the GT processor. The GT processor is realized by a Spectrum Signal Processing Inc. TMS320C30 digital signal processor board and communicates with the GT processor i/f board via the DSPLINK interface through the backplane. This report includes implementation details of the clock generation and interface circuitry and a user's guide for the proper configuration, installation and operation of the GT processor i/f board.

Résumé

Un simulateur d'un terminal au sol a été developé au Centre de Recherches pour la Défense à Ottawa (CRDO). Ce simulateur fait partie du travail au CRDO concernant les aspects de la synchronisation de la liaison sol-espace pour les communications par satellite avec spectre étalé dans la bande de fréquence extrêmement haute. Le simulateur a besoin des signaux d'horloge pour les sauts de fréquence et pour les données, et d'un interface entre le processeur et un contrôleur de synthétiseur de fréquences pour la commande et pour la transmission de données. Une carte d'interface pour le processeur du terminal au sol a été conçue et fabriquée au CRDO pour remplir ces exigences. Ce rapport décrit les fonctions de cette carte d'interface et les exigences reliées aux essais de synchronisations et à l'interface avec le contrôleur. La carte d'interface est fabriquée sur une plaquette à circuit imprimé, installée par enfichage dans le fond de panier, et contrôlée par le processeur du terminal au sol. Le processeur est réalisé par une unité de traitement de signaux numériques, TMS320C30, et communique avec la carte d'interface en passant par l'interface DSPLINK. Les détails d'implémentation de la carte d'interface sont inclus dans ce rapport ainsi qu'un guide d'utilisation.

Executive summary

Ground terminal (GT) simulator and payload (PL) simulator subsystems are being developed at Defence Research Establishment Ottawa (DREO) and Communications Research Centre (CRC) as part of the in-house work examining aspects of uplink synchronization for extremely-high-frequency (EHF) frequency hopping satellite communications (SATCOM) with on-board processing. Uplink synchronization is the process whereby a GT aligns its own clock with that of the PL (system) clock. It involves the transmission of synchronization probes by the GT at various timing offsets. The PL terminal processes the received signal and formulates a synchronization response indicating whether the probes were detected. The synchronization response is transmitted back to the GT and is used to adjust the GT clock until it is aligned with the system clock.

Part of the GT simulator includes a GT processor interface (i/f) board. This report describes the functions of the i/f board, any timing requirements related to the uplink synchronization trials, and the implementation of the functions. The GT processor i/f board generates hop clock and data clock signals and provides an interface to a hopping synthesizer controller (HSC) used in the uplink synchronization system. The i/f board is a printed circuit board with one end connected to a backplane for communication with the GT processor and the other end connected to the HSC.

The required clock signals generated by the GT processor i/f board are a hop clock and a data clock. The hop clock is a 16 kHz signal which is used by the GT processor and the HSC. The data clock is a 2.4 kHz signal which is brought back to the backplane for use by a data source. The GT processor i/f board also makes available a 4.8 kHz signal for use as a data clock. The desired data clock frequency is selected by a jumper connection.

The GT processor i/f board includes an HSC command interface and strobe generation circuit. The command interface allows the GT processor to initialize and command the HSC to various modes of operation. In addition, the GT processor is able to command the HSC to switch quickly to precomputed frequencies. The strobe generation circuit is used to produce the strobe pulse required by the HSC to read in the command to be serviced.

A data interface between the GT processor and the HSC is also provided by the i/f board. The data interface allows the GT processor to transfer the frequency-shift-keying (FSK) bin and channel number of a hop to be transmitted. A data underflow circuit is included on the i/f board to monitor whether data is provided to the HSC at every hop.

A user's guide is included in this report for the configuration, installation, and operation of the GT processor i/f board. Examples of the procedure followed for each of the functions are given with sample code written for a Spectrum Signal Processing Inc. TMS320C30 digital signal processing board.

Table of contents

List of figures

List of Tables

List of Symbols and Abbreviations

C_{ext}	External capacitance of multivibrator
CMOS	Complementary metal oxide semiconductor
CRC	Communications Research Centre
D/A	Digital-to-analog
DREO	Defence Research Establishment Ottawa
DSP	Digital signal processor
DSPLINK	50-pin DSP I/O expansion interface
EHF	Extremely high frequency
f_{base}	Base frequency parameter for the HSC
f_c	Clock frequency of the NCO
FIFO	First-in-first-out
f_o	Desired output frequency of the NCO
FSK	Frequency-shift-keying
GT	Ground terminal
HD1-HD3	Jumper header labels
HSC	Hopping synthesizer controller
i/f	Interface
I/O	Input/output
J1	Connector between GT processor i/f board and HSC command interface
J2	Connector between GT processor i/f board and HSC transmit data interface
J3	Connector between GT processor and GT processor i/f board
LSB	Least significant bit
MILSATCOM	Military satellite communications
MSB	Most significant bit
NCO	Numerically controlled oscillator
PC	Personal computer
PCB	Printed circuit board
PL	Payload
R_{ext}	External resistance of multivibrator
SATCOM	Satellite communications
TTCP	The Technical Cooperation Program
TTL	Transistor-to-transistor logic
t_w	Width of pulse generated by multivibrator
$\Delta\phi$	32-bit phase increment for the NCO

1.0 Introduction

1.1 Background

The in-house research activity at the Defence Research Establishment Ottawa (DREO) involving extremely high frequency (EHF) satellite communications (SATCOM) is continuing from the investigation of aspects of downlink synchronization and data communication [1] to those of uplink synchronization and data communication. As with the downlink synchronization experiments, the military SATCOM (MILSATCOM) groups at DREO and the Communications Research Centre (CRC) are developing the ground terminal (GT) and payload (PL) subsystems required to implement the uplink portion of an EHF processing SATCOM system employing the data link standard described in [2]. The uplink synchronization experiments are carried out over the UK Skynet 4 EHF transponder under a Memorandum of Understanding established under Subgroup S of The Technical Cooperation Program (TTCP). Subgroup S of TTCP includes participants from five nations (Australia, Canada, United Kingdom, United States, and New Zealand) who examine areas related to military satellite communications that are of interest to each country.

A system block diagram of the uplink synchronization experiments is included in Fig. 1.1. Both the GT and PL terminals are ground-based and are located about 1.5 km apart on the DREO/CRC site. The modulation scheme used in the experiments is 8-ary frequency-shift-keying (FSK). Data to be transmitted from the GT is first transferred to the GT processor i/f board. The data is then read by the HSC which calculates the frequency of the next hop. The result of the calculations is a frequency value which includes the modulated data and the hop frequency. The HSC then passes the frequency value to a frequency synthesizer. The output of the frequency synthesizer is converted to the RF transmit signal. The RF signal is transmitted at EHF to the satellite, which translates and retransmits the signal at X-band to the PL terminal. The received signal on the PL side is downconverted and transferred to a frequency synthesizer which is controlled by another HSC. The frequency synthesizer produces the hopping pattern used to dehop the received signal. The dehopped signal is then demodulated by a Fast Fourier Transform (FFT)-based block demodulator [3]. The demodulated data is stored on a First-in-first-out (FIFO) interface board until the PL processor is ready to process the data. The entire process described above and illustrated in Fig. 1.1 represents only the uplink portion of an actual SATCOM system with on-board processing. In an actual system, once the data is processed, a response is generated and transmitted back to the GT on the downlink. For the uplink synchronization trials, the downlink portion is simulated using a direct serial link which will be used to transmit the responses generated by the PL processor to the GT. The data source and sink shown in Fig. 1.1 are used in the experiments for data communications once synchronization has been achieved.

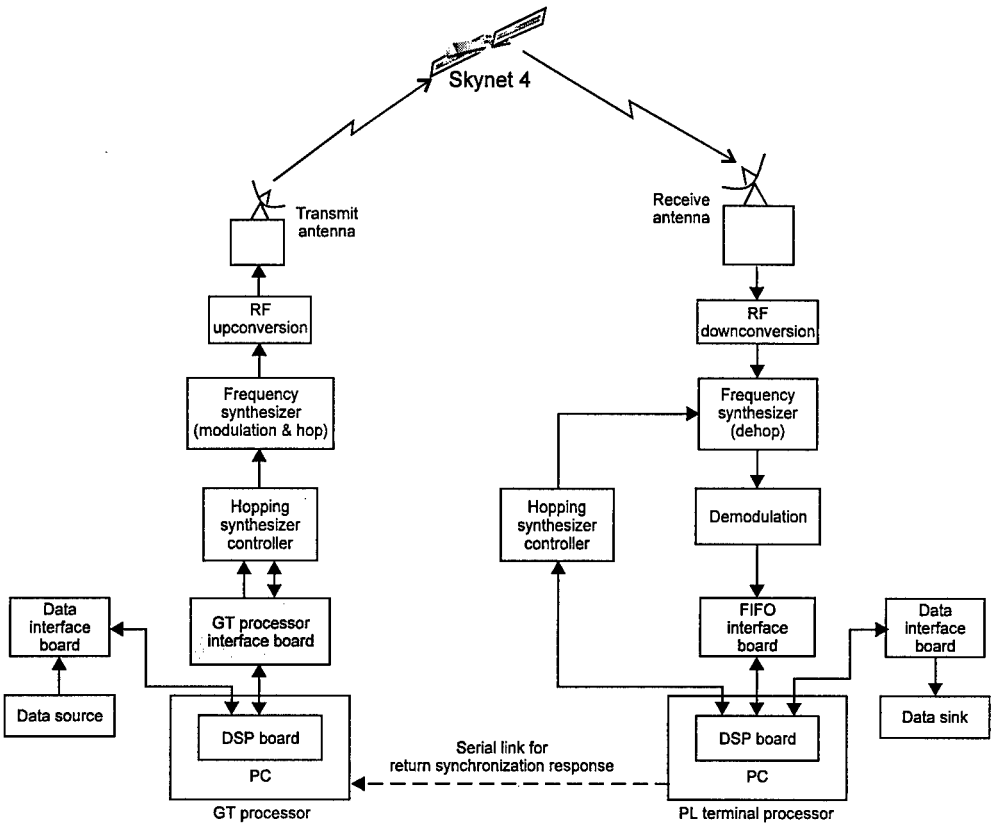


Fig. 1.1 Overall system block diagram of the uplink synchronization experiments

1.2 Description of uplink synchronization process

In uplink synchronization, the GT simulator is attempting to synchronize its hop clock to that of the payload simulator which is the master clock. The GT simulator accomplishes this by transmitting bursts of synchronization probes corresponding to different timing offset hypotheses. The payload simulator receives and processes these synchronization probes. It then produces a response indicating whether a particular burst is detected and if so, the timing error of that burst relative to the PL clock. The response is relayed back to the GT simulator which uses the responses to align its hop clock with the payload clock.

The uplink synchronization process is carried out in two stages referred to as coarse and fine synchronization. Coarse synchronization is achieved when the PL detects the GT's synchronization probes. For the uplink synchronization trials, coarse synchronization occurs when the GT simulator aligns its clock to within half a hop of the payload clock. During fine synchronization, the GT refines the alignment of the clock to within 10% of a hop.

From the data link standard described in [2], the format for transmitting synchronization probes and data is defined in terms of a time-frequency plan. The general structure of the plan is illustrated in Fig. 1.2. Along the time axis, synchronization probes and data are transmitted according to a frame/subframe structure which is repeated on a frame basis. In addition to being transmitted at a specific time, synchronization probes and data are assigned to specific channels. Each channel is subdivided into 8 frequency tone bins for 8-ary FSK.

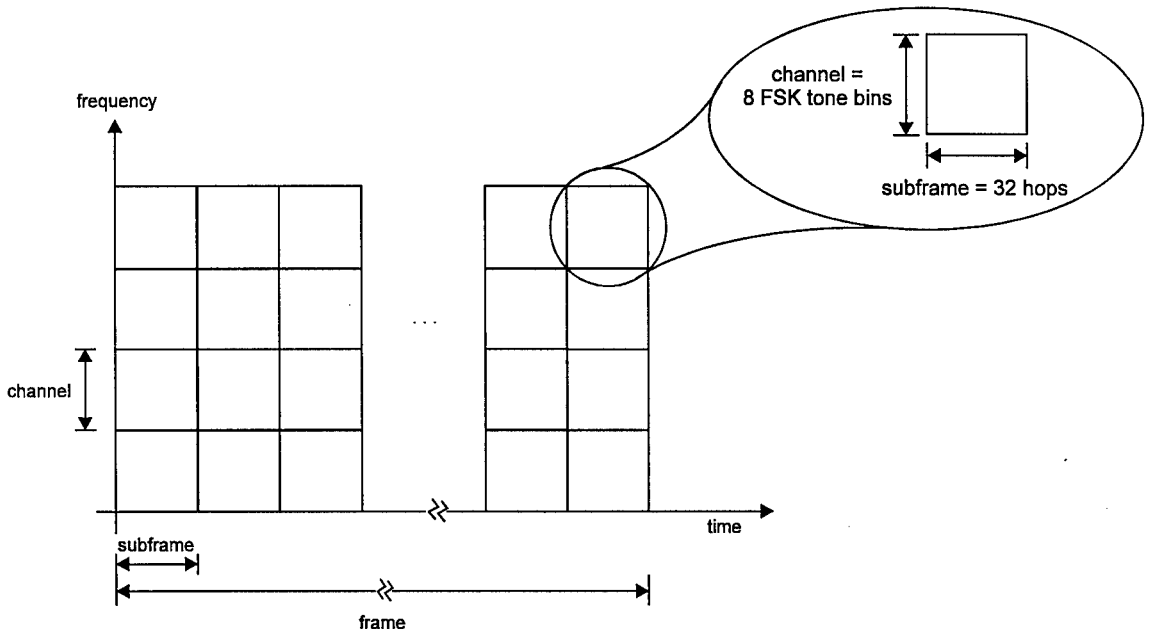


Fig. 1.2 General time-frequency plan for synchronization probe and data transmission

Coarse synchronization probes are transmitted in a specific subframe and are composed of two bursts of sixteen hops per burst [2]. Furthermore, the bursts of synchronization probes are to be transmitted in a specified channel and FSK tone bin. The coarse synchronization probes are produced by a frequency synthesizer which is controlled by a hopping synthesizer controller (HSC). The GT processor transmits the coarse synchronization probes by issuing a command to the HSC to precompute the probe frequencies for the two bursts and by commanding the HSC to switch to those frequencies at an appropriate time. The response generated by the payload simulator during coarse synchronization is a binary response, i.e. it is either a detect or a no-detect. Fig. 1.3 shows the general data flow for coarse synchronization.

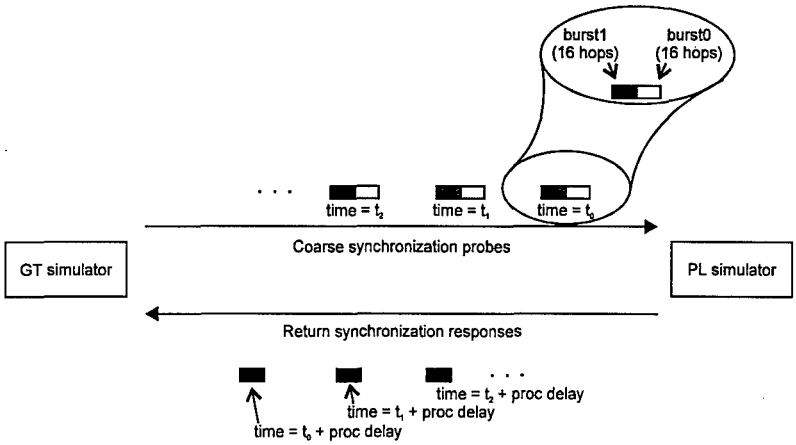


Fig. 1.3 Data flow of the coarse synchronization process

During fine synchronization, probes are also transmitted in a specific channel and FSK tone bin. They are generated in single bursts of 32 hops by the frequency synthesizer. The frequency tone bin and channel number of the fine synchronization probes are transferred at the appropriate time (subframe) by the GT processor to the HSC. The payload simulator response during fine synchronization is an estimate of how early or late the received fine synchronization probes are in relation to the payload clock. The GT processor uses this response to fine-tune its hop clock. The data flow for fine synchronization is shown in Fig. 1.4

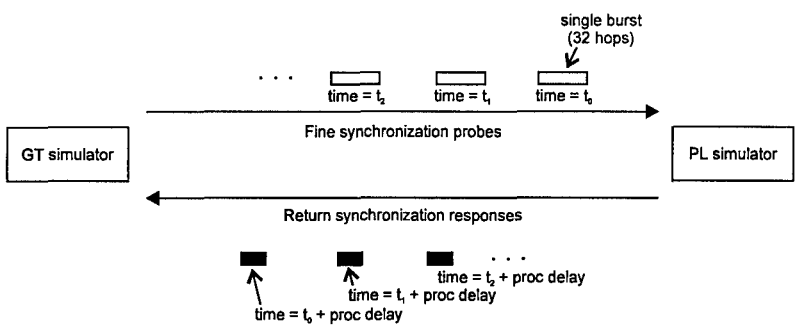


Fig. 1.4 Data flow of the fine synchronization process

1.3 Problem/Requirements

For the GT simulator in the uplink synchronization experiments, GT clock signals are required. The first is a hop clock for which synchronization with the payload hop clock is to be achieved prior to communication. To support the fine synchronization process during which the hop clock is fine-tuned, there is an additional requirement for the hop clock signal to be adjustable. The second clock signal is a data clock to be used

by a data source to transfer data to the GT processor for transmission. Furthermore, as described in Section 1.2, the GT processor drives an HSC. The HSC was also developed in-house at DREO, used in the downlink synchronization experiments, and is well documented in [4]. The requirements for driving the HSC include a command interface and a transmit data formatter. It was decided that the functions or requirements described above could all be implemented on one interface (i/f) board. This board is called the GT processor i/f board.

1.4 Objectives/Format of the report

The purpose of this report is to describe the functions of the GT processor i/f board and their implementation. This report also includes a user's guide which provides details on how to properly install, configure, and operate the GT processor i/f board. The report is organized as follows: Section 2 gives a general description of each of the functions of the GT processor i/f board and any requirements relating to the uplink synchronization experiments or timing. Section 3 describes the circuit implementation and operation of each of the functions and also describes the external interfaces. Section 4 summarizes the content of the report. Appendix A contains a user's guide on installing, configuring and operating the GT processor i/f board. The latter is shown via examples of each of the operations supported by sample code written for a Spectrum Signal Processing Inc. TMS320C30 digital signal processing (DSP) board. Another appendix includes timing diagrams obtained during testing and verification of the GT processor i/f board and printed from a logic analyser display.

2.0 Functions of the ground terminal processor interface board

2.1 General

From the description of the overall uplink synchronization process given in Section 1.2 and the resulting requirements listed in Section 1.3, the GT processor i/f board performs three principal functions as highlighted in Fig. 2.1. The first function is to generate the necessary clock signals for the processor and for the data source. This function also includes generating a periodic interrupt to the GT processor to update counters and to perform necessary operations for the synchronization process. The second function of the i/f board is to provide the circuitry for commanding the HSC. The third function is to provide the circuitry required to transfer the data to be transmitted to the HSC. In addition to the three functions required for uplink synchronization the i/f board contains the interface circuitry between the GT processor and the GT processor i/f board. Command and status registers are also included on the i/f board to set up the functions and to monitor the operations on the board. The following paragraphs describe the implementation of each of the blocks shown in Fig. 2.1.

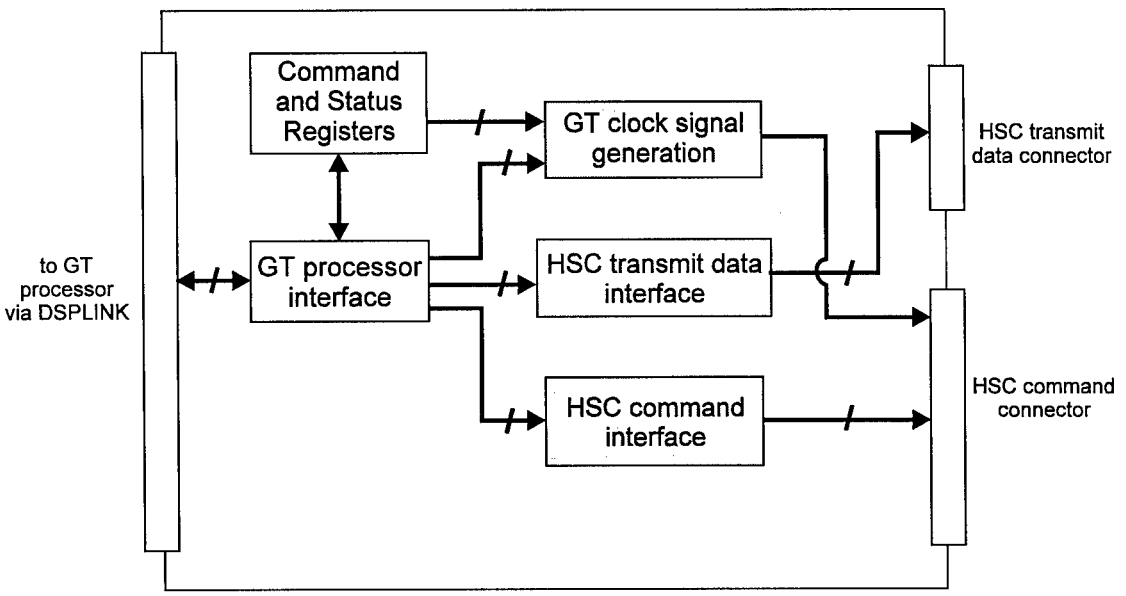


Fig. 2.1 General block diagram of GT processor i/f board functions

2.2 Description details and implementation

2.2.1 Clock generation

The first function of the GT processor i/f board is to provide the necessary clock signals for the uplink synchronization experiments. There are two clock signals which are generated for the experiments: a hop clock signal and a data clock signal. The hop clock is a 16 kHz, 50% duty cycle square wave that is provided to the HSC for its operations, and is used to generate the interrupt signal to the GT processor. It is also used in the data underflow circuit to ensure that the user data is transferred to the HSC by the GT processor continuously once synchronization is achieved and data is being transmitted. A data clock is generated on the GT processor i/f board for the ground terminal's data source. The data clock signal is provided to the data source via the DSP backplane interface. The data clock required according to [2] is a 2.4 kHz, 50% duty cycle square wave.

2.2.1.1 Numerically-controlled oscillator and D/A converter

In order to provide maximum flexibility for tuning the frequency of the hop clock signal in response to synchronization requirements, a numerically-controlled oscillator (NCO) is used. The NCO generates a fine-resolution, digitized sine wave by using a 32-bit phase increment register which is loaded by the GT processor into the NCO via the i/f board. During fine synchronization, the output frequency of the NCO can be changed by loading a new 32-bit phase increment into the NCO. In conjunction with a digital-to-analog (D/A) converter, the result is an analog signal with the desired frequency. The process of initializing the phase increment register of the NCO to produce the desired frequency is discussed in Section 3.2.2 and in Appendix A, Section A3.1. The analog waveform is transformed to the required square waveform using a comparator and appropriate divider circuit discussed in the next paragraph.

2.2.1.2 Comparator and divider circuit

A comparator is used to convert an analog signal from a D/A converter into a square wave. For the GT processor i/f board, it was decided that the output of the comparator would be a common integral multiple of both the hop clock and the data clock for maximum flexibility and to ensure the clock signals were synchronized with each other. A frequency of 192 kHz was chosen as the common multiple and allows for possible expansion of clock signal generation. The output of the comparator is divided by 12 to produce the hop clock signal (16 kHz), and is divided by 80 to produce the data clock (2.4 kHz). The final divider circuit implemented is presented in Section 3.2.2.

2.2.2 Hopping synthesizer controller command interface

A frequency synthesizer is used in the uplink synchronization trials to generate the appropriate tone to be transmitted by the ground terminal. This synthesizer is driven by an HSC developed at DREO [4]. The HSC, in turn, is controlled by the GT processor through the GT processor i/f board. The HSC responds to commands issued by the GT processor related to initialization of the HSC and to the operations required for each step of the synchronization process. The latter include the computation of synchronization probe frequencies and the rapid switching of these frequencies on demand. In general, the commands are formatted as 16-bit words. The HSC parameters are formatted as 32-bit words and are transferred to the HSC as two 16-bit words. Both a command and its associated parameter, if any, are transferred to the HSC using a single interface circuit. Details of the interface are given in Section 3.2.4. A complete list of HSC commands and their respective parameters is given in [4].

2.2.2.1 Strobe signal

When an HSC command and its parameter are issued by the GT processor, they are first transferred to a latch on the GT processor i/f board 16 bits at a time. After each 16-bit word is transferred, a strobe signal is then required to load the data from the latch to the command register of the HSC [6]. The strobe signal is generated on the GT processor i/f board after each of the 16-bit words is transferred to the latch. When the data is transferred to the HSC on the rising edge of the strobe signal, the HSC generates a "1" on its NReady line to indicate it is processing the data. The HSC is finished processing the data when the NReady line is returned to a "0". Fig. 2.2 shows the timing requirements of the strobe signal as defined in [6].

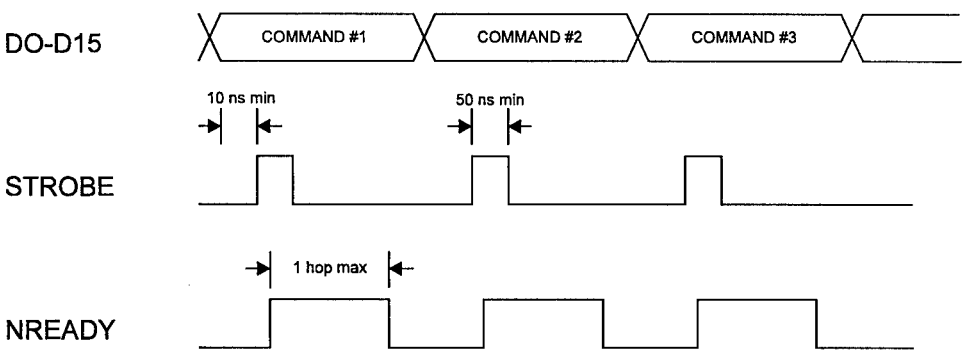


Fig. 2.2 Hopping synthesizer controller command timing specification

2.2.3 Hopping synthesizer controller transmit data interface

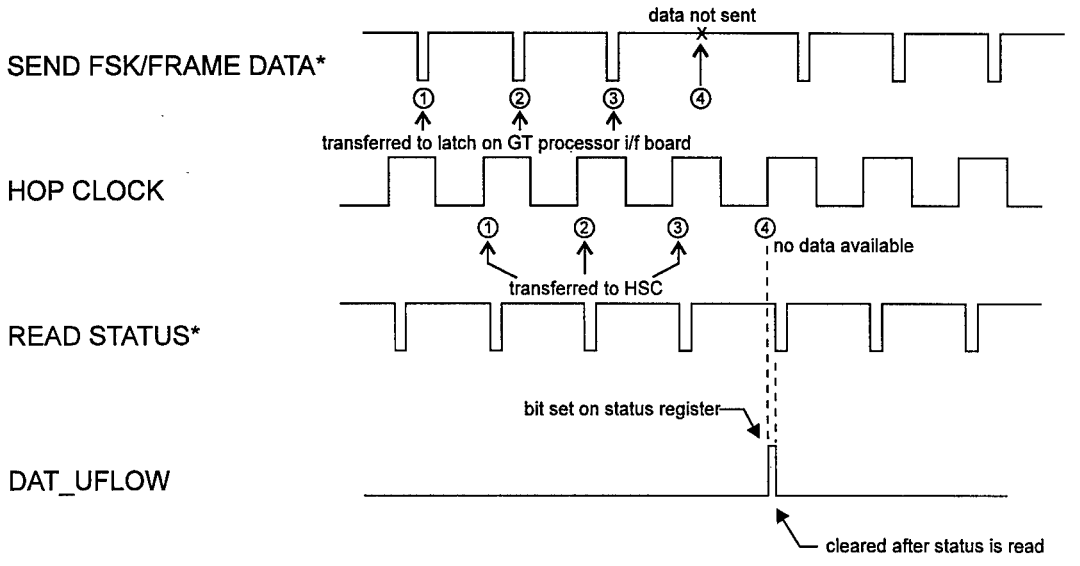
2.2.3.1 Transmit data

The data link standard [2] specifies that the allocated bandwidth of the system is divided into 32 channels and that transmitted data is formatted in a frame structure as described in Section 1.2. For the uplink synchronization trials, the modulation scheme used for synchronization probe and data transmission is 8-ary FSK. A user is assigned a specific channel and time slot during which it is permitted to transmit data. As the application is a frequency-hopped communications system, the HSC computes the actual hop frequency to be transmitted after applying an appropriate pseudorandom number to the user data. The information required by the HSC from the GT processor is the FSK tone number (3 bits) and the assigned channel number (5 bits) for a particular user. This information is transferred from the GT processor to a latch located on the GT processor i/f board. On the rising edge of the hop clock, the HSC reads the data from the latch and processes it to produce the next hop frequency. Details on the bit allocation of the FSK tone and the channel number for the latch are provided in Section 3.2.5.

When no data is being transmitted, the GT processor can effectively shut off the transmitter by setting the RF_OFF bit (active-high) on the HSC data interface to a "1". When the RF_OFF bit is set, the HSC disables the RF output of the frequency synthesizer.

2.2.3.2 Data underflow indicator

When the HSC is operating in RUN mode, the transmit data, also referred to as the FSK/frame data, must be available to be read by the HSC at the beginning of every hop. This is accomplished by writing the data for the next hop to the data latch connected to the FSK/Frame I/O port of the GT processor i/f board. At every hop, the GT processor reads the status and checks the data underflow bit (see Section 2.2.5). If the data for the hop is not present on the latch when the HSC reads the data, an underflow signal, called DAT_UFLOW, is generated and is reflected on the status register (bit D3). This may occur if the GT processor is overloaded by its operations and thus is not able to transfer the required data at the appropriate time. Under these circumstances, the GT processor will observe, upon reading the status register, the underflow condition and an appropriate error message will be sent to the user. An example of the sequence of events leading to an underflow condition is illustrated in Fig. 2.3. During the first three hops, data for the following hop period is transferred to the GT processor i/f board after the status register is read, as indicated by the SEND FSK/FRAME DATA* signal. On the fourth hop, data is not transferred to the GT processor i/f board. This causes the DAT_UFLOW bit of the status register bit to be set on the rising edge of the following hop clock. When the status register is read during that hop and the DAT_UFLOW bit is checked, the underflow condition will be detected. It is noted that the DAT_UFLOW bit is cleared upon reading the status register.



*denotes active-low signal

Fig. 2.3 Example of underflow condition

2.2.4 DSPLINK interface

The GT processor functions have been implemented on a Spectrum Signal Processing Inc. DSP board using a TMS320C30 DSP chip. The DSP board is contained in a personal computer (PC). The communication between the GT processor and the GT processor i/f board is effected by way of the DSPLINK interface on the DSP board. DSPLINK is a high-speed 16-bit bi-directional parallel input-output (I/O) data bus interface [5]. The DSPLINK interface is available on all of Spectrum's DSP boards and thus, does not limit the potential use of the GT processor i/f board with other Spectrum DSP boards.

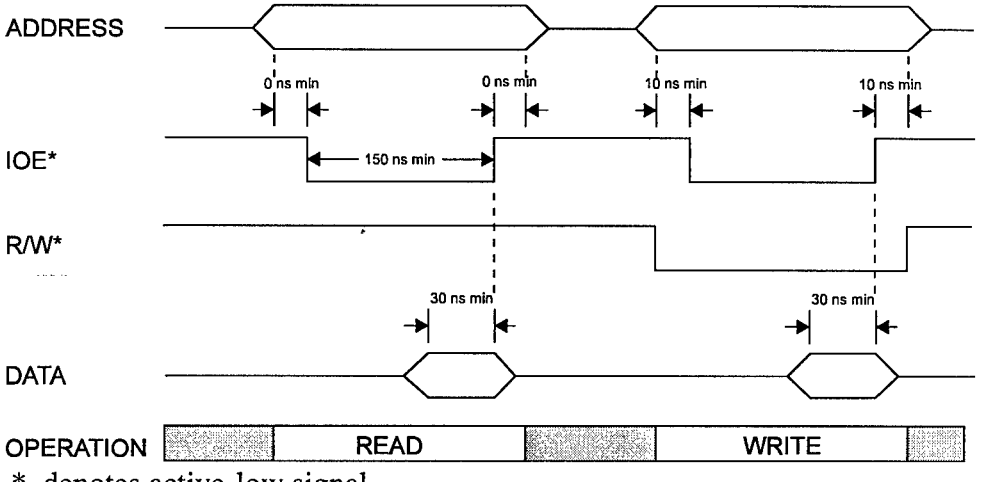
The DSPLINK interface includes many signals which are used to communicate with the DSP board. However, for the GT processor i/f board application, a subset of those signals is used. The subset is listed in Table 2.1.

SIGNAL NAME	IN/OUT	DESCRIPTION
D0-D15	I/O	16 fully buffered bi-directional data lines
A0-A3	O	4 buffered address lines
R/W*	O	A READ/WRITE* line to indicate direction of data transfer. A WRITE refers to a transfer from the master board to the slave board.
IOE*	O	I/O enable line. Indicates access to one of 16 standard I/O ports only
INT0*	I	Negative-edge, or low level interrupt
RESET*	O	Reset line, same as reset into processor
FLAGOUT	O	Single-bit general purpose output line
GND	--	Signal ground

* denotes an active low signal

Table 2.1 DSPLINK signal subset used for the GT processor i/f board

For a general purpose I/O access using DSPLINK, the timing requirements for a read and write operation [5] are presented in Fig. 2.4.



* denotes active-low signal

Fig. 2.4 DSPLINK timing specifications for read and write accesses

2.2.4.1 DSP backplane

For the uplink synchronization experiments, the GT processor communicates with several components of the GT simulator via the DSPLINK interface. In order to accommodate other custom boards which use this interface to communicate with the GT processor, a DSP backplane containing the DSPLINK interface was implemented to minimize the number of ribbon cables needed. In addition, the backplane provided a convenient way for custom boards to communicate with each other, thereby reducing again the number of cables. The GT processor i/f board incorporates the backplane as the external interface to the GT processor. The DSP backplane pinout is described further in Section 3.2.6

2.2.4.2 GT processor i/f board input/output ports

With four address lines available, sixteen possible I/O ports can be accessed using only the DSPLINK signal subset. The GT processor i/f board requires a total of four output and two input I/O ports to realize the functions described above. In this implementation, the output direction flows from the DSP to the i/f board. Conversely, an input port refers to a process which goes from the i/f board to the DSP. A description of the GT processor i/f board I/O ports is given in Table 2.2.

NAME	I/O	DESCRIPTION
COMMAND	O	Command register port. Used for software reset of the GT processor i/f board as well as for commanding the numerically-controlled oscillator (NCO). See Section 2.2.5.
STATUS	I	Status register port. See Section 2.2.5.
NCO	O	NCO port. Used to load phase values into the NCO to generate the required clock signals. See Sections 2.2.1.1 and 3.2.2.
INTRPT	I	Interrupt clear port. Used to clear interrupt signal to the DSP.
HSC	O	HSC command port. Used to issue commands to the HSC.
FSK/FRAME	O	HSC data interface port. Used to specify FSK data and Channel number for transmission by the hopping synthesizer [3].

Table 2.2 GT processor i/f board input/output ports

2.2.4.2.1 Base addressing scheme

As mentioned above, there are sixteen possible I/O ports available on the DSPLINK interface. In order to allow for maximum flexibility in assigning addresses to the custom boards which use DSPLINK, and to avoid any addressing conflicts between these same boards, a base addressing scheme is used. A base addressing scheme consists of utilizing the minimum number of address lines to generate the required number of I/O ports for a particular board. The remaining address lines are used to define a base address so that the sixteen available I/O ports are effectively partitioned to accommodate the various boards using the DSPLINK interface bus.

Using the base addressing scheme, the GT processor i/f board requires two address lines in conjunction with the R/W* line of DSPLINK to generate the appropriate I/O ports. The two remaining address lines determine the base address of the GT processor i/f board. The resulting address assignments and the possible base addresses for the GT processor i/f board are listed in Table 2.3 and Table 2.4 respectively. For the GT processor developed on a TMS320C30 DSP board for the uplink synchronization experiments, the I/O ports for DSPLINK are implemented as memory-mapped registers starting at address 800000 h (hexadecimal) of the DSP memory. The last four bits of the 24-bit address are the address lines A0-A3 on DSPLINK with A0 being the least significant bit (LSB). In order to write data to an I/O port, the data is stored in the corresponding memory-mapped register. Reading data from an I/O port is done by loading the contents of the memory-mapped register into a register to be used by the DSP board.

I/O PORT	A1	A0	R/W*	RESULTING ADDRESS
COMMAND	0	0	0	base address + 0
STATUS	0	0	1	base address + 0
NCO	0	1	0	base address + 1
INTRPT	0	1	1	base address + 1
HSC	1	0	0	base address + 2
not used	1	0	1	base address + 2
FSK/FRAME	1	1	0	base address + 3
not used	1	1	1	base address + 3

* denotes active low signal

Table 2.3 GT processor i/f board I/O port address assignments

A3	A2	BASE ADDRESS
0	0	0 h
0	1	4 h
1	0	8 h
1	1	C h

Table 2.4 GT processor i/f board base addresses

2.2.5 Command and status register

The command register of the GT processor i/f board allows the user to perform a software reset of the board as well as to direct the NCO to load the appropriate phase values to generate the clock signals. The command register for the GT processor i/f board is configured as shown in Fig. 2.5 and should be initialized to its default value, 0005 h. By writing a "0" or a "1" to the appropriate bit in the command register, a "low" or a "high" level signal is generated on the GT processor i/f board respectively. The software reset signal (D2) is an active-low signal. When the software reset signal is issued to the GT processor i/f board, circuitry (e.g. latches, dividers) is reset to its default settings. Data bits D0 and D1 of the command register are used to generate an active-low WRN pulse and an active-high LDSTB pulse for the NCO. These signals are further described in Section 3.2.2. The command register must always be reset to its default value once the desired signal has been generated.

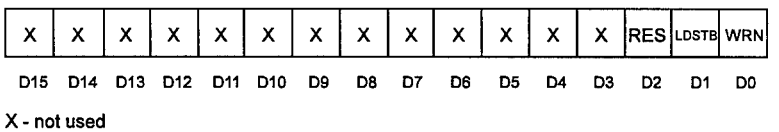


Fig. 2.5 GT processor i/f board command register

Similarly, the status register consists of assigned bits which can be read from the GT processor i/f board to monitor the status of various operations. The status register can be read at any time during the operation of the i/f board. Fig. 2.6. shows the status bit assignment for the GT processor i/f board. Bit D0 is the hop clock signal of the GT processor i/f board. The hop clock signal is included in the status register to enable the GT processor to monitor the signal. Bit D1 is a SYNC line which originates from the HSC. The SYNC line is an active-high which indicates that the HSC is ready to respond to commands to switch quickly to precomputed frequencies during coarse synchronization. The NRDY bit of the status register (D2) represents the NReady line, also from the HSC. The NReady line is an active-low line which signals when the HSC has read the previous data from the HSC command interface latch and is ready for the next data. The user must check that this line is low before transferring data to the HSC command latch. Bit D3 of the status register is the data underflow bit from the HSC transmit data interface.

The GT processor checks this bit to ensure that transmit data is provided continuously to the HSC during data communications (see Section 2.2.3.2).

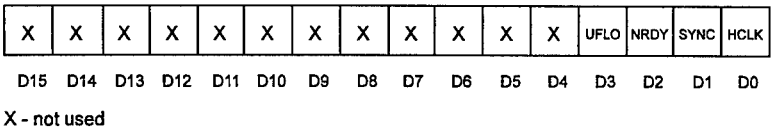


Fig. 2.6 GT processor interface board status register

2.2.6 Interrupt generation

The operations of the synchronization process occur on a hop basis. These include the updating of counters, transmission of synchronization probes and data, adjusting of clock frequencies, and commanding the HSC. As a result, the GT processor i/f board generates an active-low interrupt signal to the DSP on every hop clock for the GT processor to perform the necessary actions and clear the interrupt. The interrupt signal is transmitted to the GT processor via the INT0* line of DSPLINK. The INT0* line is cleared by the GT processor by performing a read operation on the INTRPT port of the GT processor i/f board. The contents read from the INTRPT port are discarded. Thus, the INTRPT register is defined as follows:

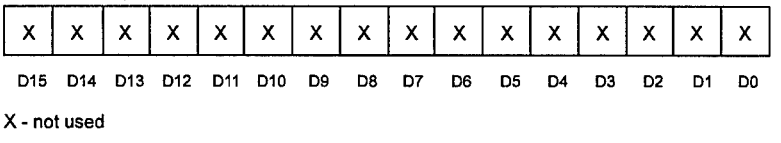


Fig. 2.7 GT processor i/f board interrupt register

3.0 Hardware details

3.1 Overall layout

The GT processor i/f board is implemented on a double-sided printed circuit board (PCB). The overall layout of the board is shown in Fig. 3.1. The components used for the i/f board are all off-the-shelf components. The schematic of the GT processor i/f board is included in Fig. 3.2. Table 3.1 lists the components of the i/f board.

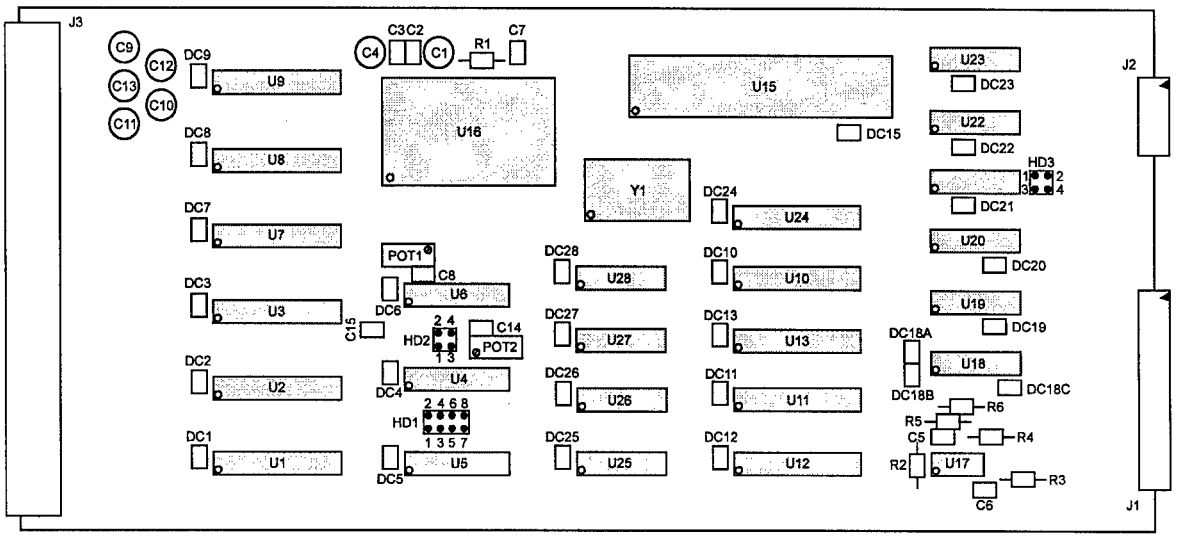


Fig. 3.1 Overall layout of GT processor i/f board

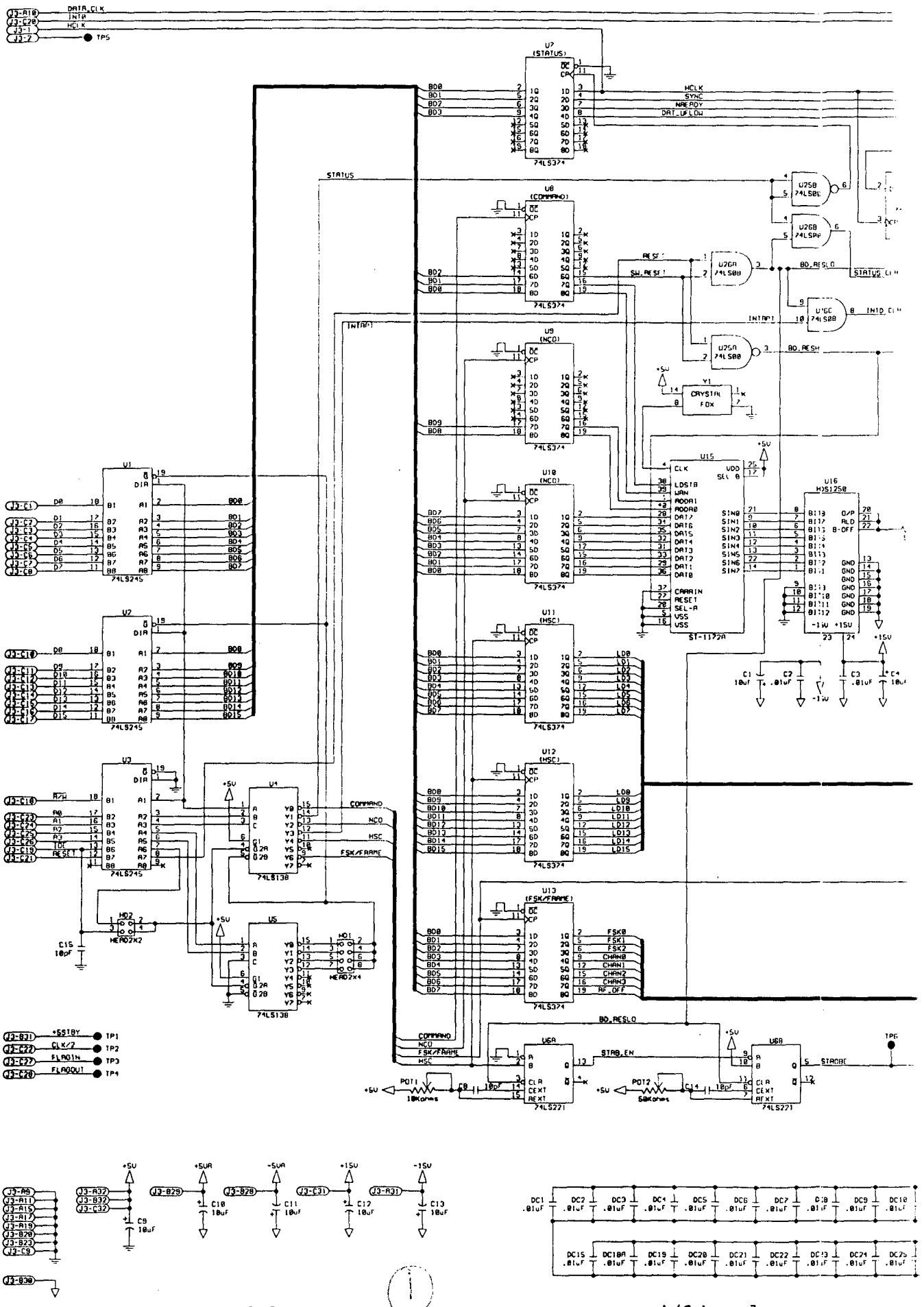
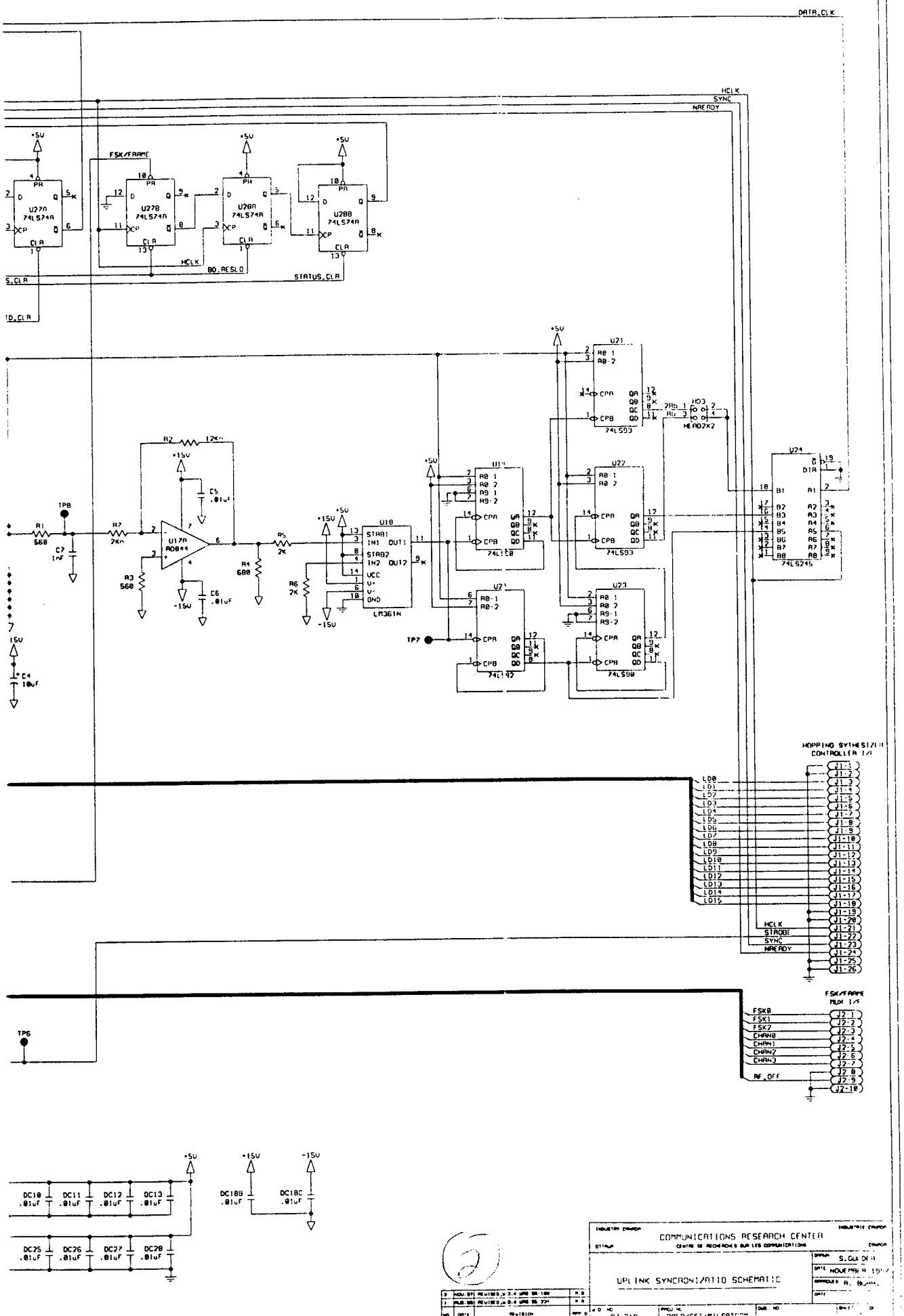


Fig. 3.2 Circuit Schematic of GT processor i/f board


DATA.CLK
HCLK
SYNC
NREADY
FSK/FRAME
U27A 74LS74A
U27B 74LS74A
U28A 74LS74A
U28B 74LS74A
BD.RESLO
STATUS.CLR
S.CLR
TD.CLR
U17A AD844
U18 LM361N
U21
U22
74LS93
74LS245
HEAD2X2
TP8
TP7
TP6
HOPPING SYTHESIZER CONTROLLER I/F
FSK/FRAME MUX I/F
STROBE
RF.OFF
COMMUNICATIONS RESEARCH CENTER
UPLINK SYNCRONIZATIO SCHEMATIC
S.GLADER
DRED/SS1/MILSATCOM

INTEGRATED CIRCUITS		
LABEL	**NAME**	**DESCRIPTION**
U1,U2,U3,U24	74LS245	Octal bus transceiver
U4, U5	74LS138	3-line to 8-line decoder/demultiplexer
U6	74LS221	Dual monostable multivibrator
U7 to U13 (incl)	74LS374	Octal D-type transparent latch
U15	STEL-1172A	50MHz, 32-bit CMOS NCO
U16	HDS1250	Ultra-high speed 12-bit D/A converter
U17	AD844	60MHz, 2000V/μs monolithic opamp
U18	LM361	High speed differential comparator
U19,U23	74LS90	Decade counter
U20	74LS92	Divide-by-twelve counter
U21,U22	74LS93	Binary counter
U25	74LS00	Quad 2-input positive NAND gates
U26	74LS08	Quad 2-input positive AND gates
U27,U28	74LS74	Dual D-type positive edge-triggered flip-flops

DISCRETE COMPONENTS	
LABEL	**DESCRIPTION**
R1,R3	560Ω, 1/4 W
R2	12kΩ, 1/4 W
R4	680Ω, 1/4 W
R5,R6,R7	2kΩ, 1/4 W
DC1-DC28	0.01μF
C1,C4,C9,C10,C11,C12,C13	10 μF
C2,C3,C5,C6	0.01μF
C7	1nF
C8,C14,C15	10pF
POT1	10kΩ potentiometer
POT2	50kΩ potentiometer
HD1	4 terminal jumper header
HD2,HD3	2 terminal jumper header

OTHER COMPONENTS	
LABEL	**DESCRIPTION**
J1	26-pin male connector, double row, 0.1" spacing
J2	10-pin male connector, double row, 0.1" spacing
J3	96-pin male connector, triple row, 0.1" spacing

Table 3.1 Interface board components list

3.2 Implementation of GT processor i/f board functions

Block diagrams of each of the major functions of the GT processor i/f board are presented in the paragraphs which follow. Descriptions of the circuit operations and any set up requirements are also provided.

3.2.1 DSPLINK interface circuit

Using a subset of the DSPLINK interface signals as described in Section 2.2.4, the I/O ports addresses are decoded according to the block diagram shown in Fig. 3.3. All the signals from DSPLINK coming onto the board are buffered. In the circuit implementation, two decoders are used. One is used to select or enable an I/O port to perform a data transfer to or from the board. The other is used to allow the user to select the base address of the i/f board. The base address of the board is determined by the location of a single jumper applied to HD1.

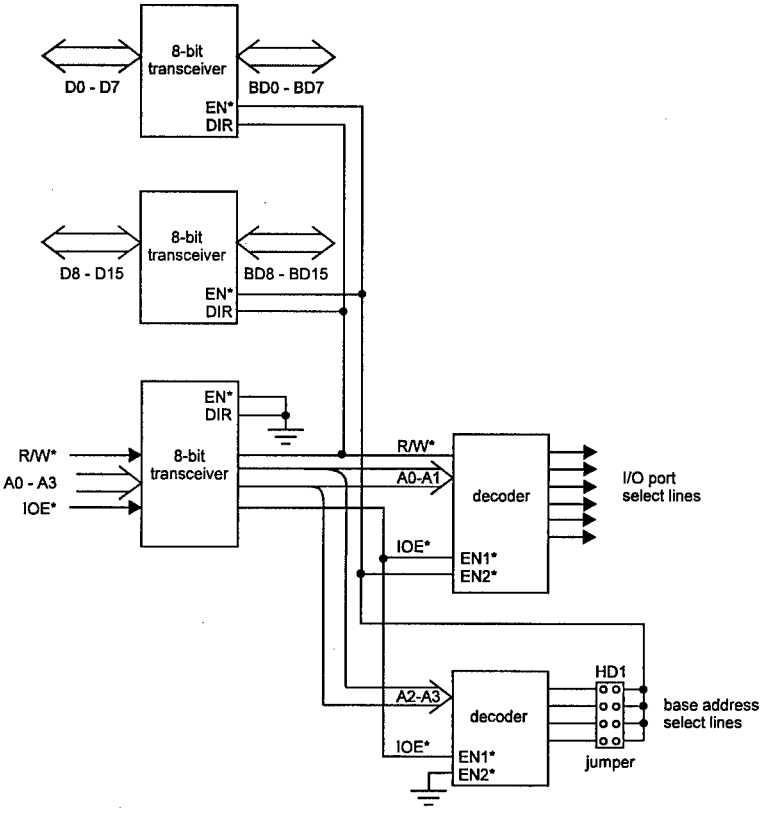


Fig. 3.3 DSPLINK address decoder block diagram

3.2.2 Clock generation circuit

The generation of required GT clock signals is described in two parts. The first part describes the generation of an analog sine wave of a specific frequency and the conversion to a square wave. The second part describes the divider circuit used to derive the required hop clock and data clock signals. The two block diagrams illustrating the operations are presented in Fig. 3.4 and Fig. 3.5 respectively.

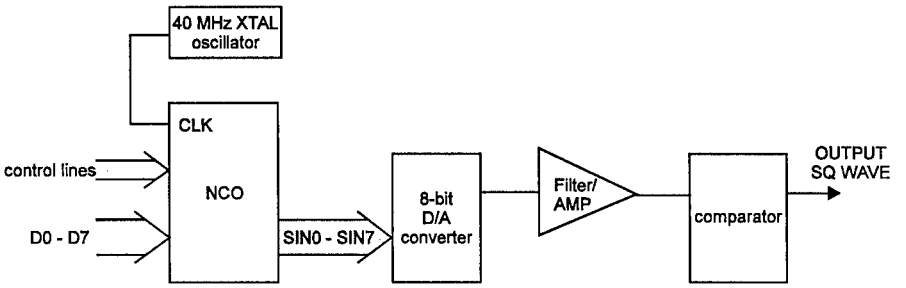


Fig. 3.4 Block diagram of the frequency generation circuit

The NCO used in Fig. 3.4. is the STEL-1172A. The NCO produces a 8-bit resolution digitally synthesized sine wave. The frequency of sine wave generated is determined by a 32-bit phase value initialized by the user according to the following formula [8]:

$$f_o = \frac{f_c \times \Delta\phi}{2^{32}} \tag{1}$$

where f_o is the desired output frequency, and f_c is the clock frequency. For the GT processor i/f board, the clock frequency is 40MHz. The 32-bit phase value, $\Delta\phi$, represents the phase increment for each cycle of the clock. The constant, 2^{32}, in Equation (1) represents the normalization factor of $\Delta\phi$. The NCO uses four control lines: ADDR0; ADDR1; WRN; and LDSTB. The ADDR0 and ADDR1 lines identify one of four 8-bit registers in the NCO used to store the 32-bit phase increment. The WRN and LDSTB lines are used to load the data into the registers and to trigger the changing of the NCO output frequency. The phase increment is entered by writing four consecutive 8-bit words to the NCO through the NCO and COMMAND ports of the GT processor i/f board. Each 8-bit word is identified by the ADDR0 and ADDR1 lines and must be specified when the user transfers the data value via the NCO port. The 8-bit word is then loaded into the NCO phase register by issuing an active-low WRN pulse via the COMMAND port. When the entire phase increment has been transferred, an active-high LDSTB pulse is issued to the NCO through the COMMAND register to execute the change in the NCO output frequency. An example of the NCO frequency setup operation is provided in the user's

guide in Appendix A. At the output of the D/A converter, HDS1250, a combination low-pass filter/amplifier is inserted to ensure that a smooth and sufficient signal level is being presented to the comparator. On the GT processor i/f board the required gain of the amplifier was found to be 6x. The comparator used on the GT processor i/f board is an LM361N. The LM361N voltage comparator takes a bipolar analog sine wave and produces a TTL square wave of the same frequency.

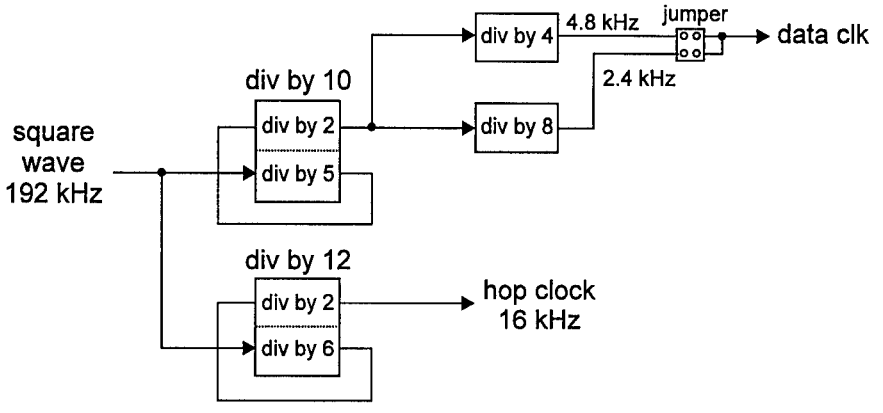


Fig. 3.5 Block diagram of the clock signals divider circuit

As described in Section 2.2.1.2, the frequency of the comparator output square wave was selected to be 192 kHz. This frequency is a multiple integral of both the hop and data clock, and allows the flexibility for possible expansion of clock signals generated. As shown in Fig. 3.5, the hop and data clock signals are generated by dividing the output signal of the comparator by 12 and 80 respectively. For the board implementation, the divide-by-12 operation is carried out using in a single integrated circuit (IC). The divide-by-80, on the other hand, is implemented in two stages. First, the square wave frequency is divided by 10. The resulting signal is then separately divided by 8 and by 4 to yield the data clock signals of 2.4 kHz and 4.8 kHz which can be selected by applying jumper to the appropriate terminal. The two separate final divider circuits provide additional flexibility in selecting alternate data clock rates.

3.2.3 Interrupt generation circuit

The active-low interrupt signal required by the GT processor is produced by a D-flip-flop as shown in Fig. 3.6. On the rising edge of the hop clock, a low signal is generated at the output which will trigger an interrupt to the GT processor DSP. The flip-flop must be cleared before the next rising edge of the hop clock occurs by performing a read operation on the INTRPT port of the GT processor i/f board. This action generates the appropriate INT0_CLR* signal for the circuit.

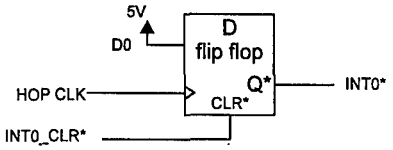


Fig. 3.6 Interrupt signal generation block diagram

3.2.4 Hopping synthesizer controller command interface circuit

The block diagram of the interface circuit between the GT processor and the HSC is presented in Fig. 3.7. A 16-bit latch circuit is used to hold the command and parameter values (i.e. the data lines) for the HSC. In addition, a STROBE signal needed by the HSC is generated by the two monostable multivibrators (aka "one-shots") after the data is latched. The first "one-shot" is used to delay the start of the STROBE to ensure that data is valid at the output of the latches before being transferred to the HSC. The second "one-shot" is used to set the width of the STROBE pulse. The width of the pulse generated by a "one-shot", t_w , is related to the values of the resistor and capacitor, R_{ext} and C_{ext}, respectively as follows:

$$t_w = C_{ext}\ R_{ext} \times \ln 2 \tag{2}$$

As specified in Section 2.2.2.1, the pulse width of the STROBE is a minimum of 50ns. To achieve 50ns, the values of C_{ext} = 10 pF and R_{ext} = 7.3kΩ are used.

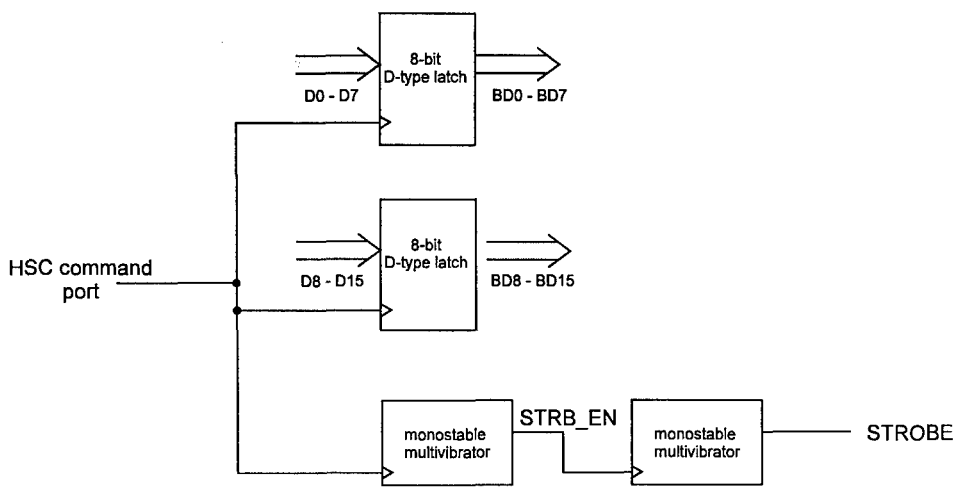


Fig. 3.7 Hopping synthesizer controller interface and strobe generation circuit block diagram

3.2.5 Hopping synthesizer controller transmit data interface circuit

The block diagram for the interface between the GT processor and the HSC transmit data interface is illustrated in Fig. 3.8. The data to be transferred are the 8-ary FSK tone number, the channel number, and the RF_OFF signal. The RF_OFF signal is an active-high signal and allows the user to effectively turn the transmitter off. This bit can be used to simulate a jamming scenario or unused hops [4]. For this implementation, it was decided that the RF_OFF signal would be part of the HSC transmit data interface. Consequently, the line representing the most significant bit (MSB) of the channel number was replaced with the RF_OFF signal so that the transmit data interface could be realized using only one 8-bit latch. The replacement of the MSB of the channel number is feasible because it was not necessary to implement all 32 channels to examine the uplink synchronization aspects of the data link standard. The bit allocation for the latch is shown in Fig. 3.9. When the FSK bin and channel number are transferred to the HSC, the MSB of the channel number input to the HSC connector is tied to ground.

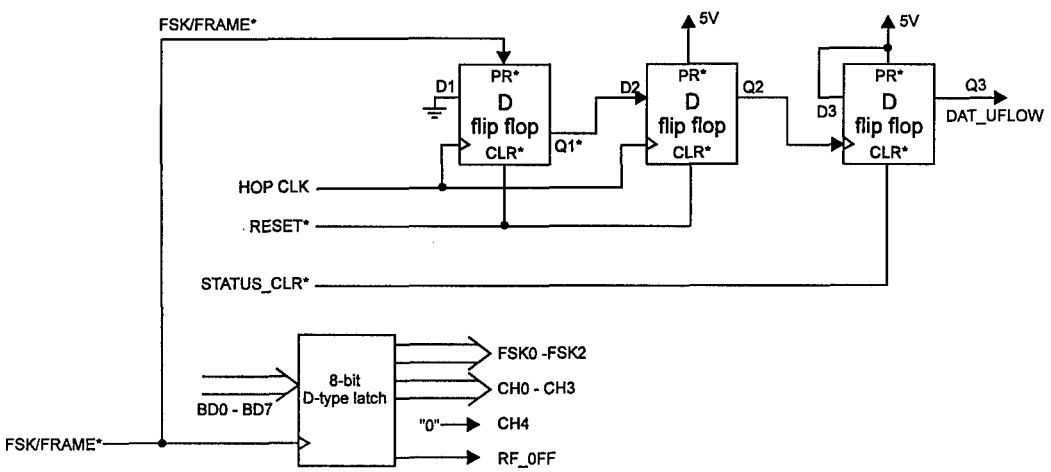


Fig. 3.8 Block diagram of the hopping synthesizer controller transmit data interface and data underflow circuit

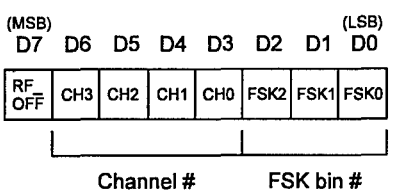


Fig. 3.9 Bit allocation for hopping synthesizer controller transmit data latch

In addition to the data to be transferred, three D-flip flops are used to implement a data underflow circuit. The underflow circuit detects when data is not provided to the HSC at every hop. Fig. 3.10 expands on the sequence of events leading to an underflow condition as first presented in Fig. 2.4. For the first three hop periods, when data is transferred to the GT processor i/f board (SEND FSK/FRAME DATA* going "low"), the output of the first flip-flop, Q1*, becomes "low". The signal Q1* is then transferred to the output of the second flip-flop before returning to a "high" on the rising edge of the following hop. If data is transferred continuously, the output of the second flip-flop, Q2, is a low signal. This is also true for the output of the third flip-flop which is the data underflow bit, DAT_UFLOW. On the fourth hop, data is not sent to the HSC. As a result, Q1* remains "high" on the rising edge of the fifth hop. This, in turn, produces a "high" signal at the output of the second flip-flop. The rising edge of Q2 causes the output of the third flip-flop to become "high", indicating the underflow condition. The underflow condition is then detected upon reading the status register on the fifth hop.

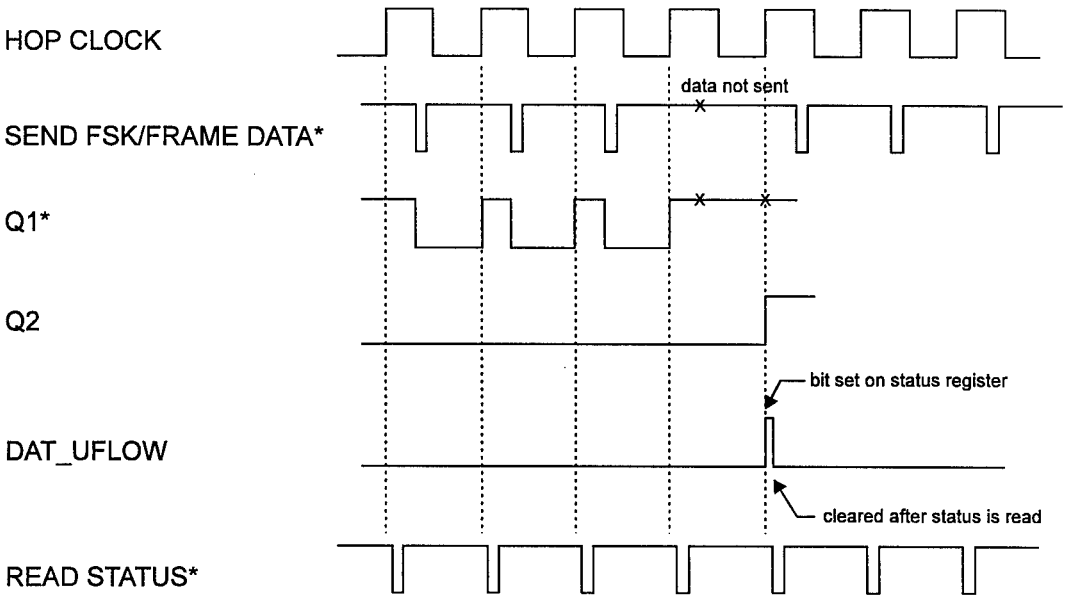


Fig. 3.10 Data underflow circuit operation

3.2.6 External interfaces

The following paragraphs describe the hardware interface connectors required by the GT processor i/f board to communicate with the GT processor (i.e. DSP board) and the HSC. A total of three (3) connectors are required for the GT processor i/f board. A description of the pin configurations is provided for each of the connectors.

3.2.6.1 DSP backplane hardware interface

The hardware interface required between the GT processor i/f board and the GT processor consists of a 96-pin connector with 0.1" spacing providing the sixteen bidirectional data lines with control signals and clock signals from DSPLINK on a backplane. The 96-pin connector is shown in Fig. 3.11. The pinout configuration for the connector is provided in Table 3.2. A brief description of the signals in the pinout configuration is also provided below.

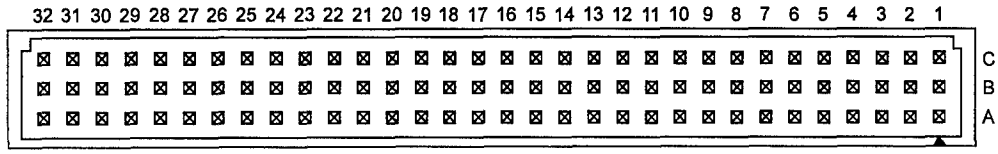


Fig. 3.11 DSPLINK interface connector (J3) pinout

PIN	ROW A	ROW B	ROW C
1	HOP CLK		D0
2	HOP CLK*		D1
3			D2
4		RESERVED	D3
5		RESERVED	D4
6		RESERVED	D5
7		RESERVED	D6
8		RESERVED	D7
9	GND	RESERVED	GND
10	DATA CLK	RESERVED	D8
11	GND	RESERVED	D9
12			D10
13			D11
14			D12
15	GND		D13
16			D14
17	GND		D15
18			W*/R
19	GND		IOE*
20	RESERVED	GND	INTO*
21	RESERVED		RESET
22	RESERVED		CLK/2
23		GND	A0
24			A1
25			A2
26			A3
27			FLAGIN
28		-5V ANALOG	FLAGOUT
29		5V ANALOG	
30		AGND	
31	-15V	5V STBY	15V
32	15V	5V	5V

Table 3.2 DSP backplane interface pinout configuration

D0-D15	Sixteen bi-directional TTL data lines of DSPLINK
GND	Digital ground
W*/R	DSPLINK read/write* line originating from the DSP to signal the direction of data transfer. The asterisk (*) denotes an active-low signal for the WRITE signal. The direction is determined from the point of view of the DSP (i.e. a WRITE refers to data being transferred from the DSP to the GT processor i/f board.
IOE*	An active-low, input/output enable signal indicating an access on the DSPLINK originating from the DSP.
INT0*	A negative-edge triggered, or active-low interrupt signal on DSPLINK generated on the GT processor i/f board.
RESET	DSPLINK reset line. This signal is active-low.
CLK/2	General purpose clock signal on DSPLINK originating from the DSP. This signal is not used by the GT processor i/f board.
A0-A3	Four buffered TTL address lines of DSPLINK.
FLAGIN	General purpose input line on DSPLINK readable by the DSP. This signal is not used by the GT processor i/f board.
FLAGOUT	General purpose output line on DSPLINK writeable by the DSP.
15V	15 volts power supply.
-15V	-15 volts power supply
5V	5 volts power supply
-5V ANALOG	-5 volts analog power supply
5V ANALOG	5 volts analog power supply
AGND	Analog ground
5V STDBY	5 volts standby power supply. This signal is not used by the GT processor i/f board.
HOP CLK	Hop clock signal originating from the GT processor i/f board.
HOP CLK*	Inverse hop clock signal. This pin is not currently being used by the GT processor i/f board.
DATA CLK	Data clock signal originating from the GT processor i/f board.
RESERVED	Reserved lines for the DSP backplane.

3.2.6.2 Hopping synthesizer controller command hardware interface

One of the hardware interfaces between the GT processor i/f board and the HSC is a 26-pin connector with 0.1" spacing. This connector provides a 16-bit unidirectional data port with handshaking and clock signals. The connector pinout is shown in Fig. 3.12. A detailed description of the signals using the pinout configuration is provided in [4]. A more general description follows:

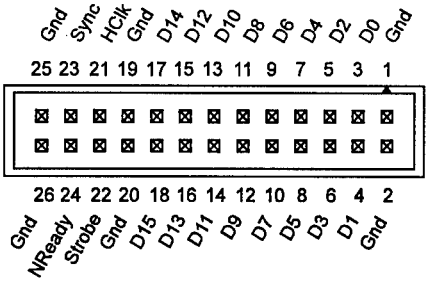


Fig. 3.12 Hopping synthesizer controller command interface connector (J1) pinout

D0-D15	Sixteen (16) TTL data lines for transferring data from the GT processor (via the GT processor i/f board) to the HSC.
GND	Digital ground
HClk	Hop clock provided by the GT processor i/f board to be used by the HSC to process commands and latch the frequency synthesizer.
NReady	An active-low signal generated by the HSC to indicate it is ready to receive the next data.
Strobe	An active-high signal from the GT processor i/f board to signal to the HSC that another command is waiting to be processed.
Sync	An active-high TTL line generated by the HSC to indicate it is in a tight loop waiting to respond to DLGO and ULGO commands for downlink coarse synchronization and uplink coarse synchronization respectively [4].

3.2.6.3 Hopping synthesizer controller transmit data hardware interface

The second hardware interface required between the GT processor i/f board and the HSC consists of a 10-pin connector with 0.1" spacing. It is used to transfer the FSK bin number (3 bits) and the channel number (5 bits) to the HSC for a particular hop period. In addition, a line

called "RF_OFF" is provided to allow the user to effectively shut off the RF transmitter and is described further in Section 2.2.3.1 and in [4]. It is noted that the GT processor only transfers 4 channel lines to the latch on the GT processor i/f board to allow the RF_OFF signal line to be implemented on the transmit data interface. As a result, the fifth channel number line, Chan4, must be tied to ground on the GT processor i/f board. Fig. 3.13. shows the pinout configuration of the connector. Again, a description of each of the lines associated with the pins is given below. A more detailed description can be found in [4].

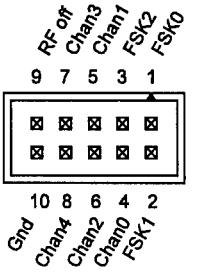


Fig. 3.13 Hopping synthesizer controller transmit data interface connector (J2) pinout

FSK0-FSK2	Three TTL data lines for transferring the FSK bin number of a particular hop to the HSC.
Chan0-Chan4	Five TTL data lines for transferring the channel number of a particular hop to the HSC. Chan4 is tied to ground on the GT processor i/f board.
RF_OFF	An active-high TTL data line used by the GT processor to effectively turn the RF power off the transmitter.
Gnd	Digital ground.

4.0 Summary

A GT processor i/f board was designed and implemented on a PCB to generate clock signals and provide the necessary interface to an HSC for EHF uplink synchronization trials carried out at DREO and CRC. The synchronization trials consist of both the coarse and fine synchronization to align the GT clock with the PL or system clock. The GT processor i/f board was designed to be driven by a Spectrum Signal Processing Inc. TMS320C30 DSP board but can be used by other DSP boards with the DSPLINK interface. The i/f board, in turn, drives an HSC, which was also designed at DREO. This report described the functions of the GT processor i/f board, any timing requirements related to the uplink synchronization trials, and the implementation of the functions.

A hop clock signal is generated on the GT processor i/f board which is used by both the GT processor and the HSC to perform appropriate operations related to the uplink synchronization process. This includes updating GT processor counters, transmitting synchronization probes and transmitting user data. The hop clock frequency generated is 16 kHz. A data clock signal is also generated for the user's data source. The data clock signal is either 2.4 kHz or 4.8 kHz, selectable by a jumper on the board.

The GT processor i/f board also allows the user (GT) to command an HSC, which in turn, drives a frequency synthesizer to transmit a particular frequency tone. This interface is used to initialize the HSC and command the HSC to precompute synchronization probe frequencies as well as to switch to these frequencies quickly.

In addition to the HSC command interface, the GT processor i/f board transfers data in the form of an FSK bin and channel number to the HSC once coarse synchronization has been achieved. To ensure that such data is provided at the beginning of every hop, a data underflow circuit is also incorporated.

A user's guide is included in Appendix A with examples and sample code of each of the functions of the GT processor i/f board. Appendix B includes printouts of a logic analyser display which verify the proper operation of the board

References

1. Addison, R.D., and Seed, W.R., "*Implementation of an EHF Frequency-Hopping Simulator*", DREO Report 1279, December 1995.
2. Lambert, J.D., "*DREO/CRC Joint Data Link Standard for Low Data Rate Service to EHF Ground Terminal Payload Simulators*", DREO Report 1069, February 1991.
3. Tom, C., and Meng, Z., "*Multichannel M-ary Frequency-shift-keying Block Demodulator Implementation*", DREO Report 1307, December 1996.
4. Addison, R., "*Modified Hopping Synthesizer Controller*", DREO Report 1304, December 1996.
5. *TMS320C30 System Board User's Manual*, Spectrum Signal Processing Inc., Issue 1.01, August 1990.
6. Addison, R., "*Hopping Synthesizer Controller*", DREO Technical Note 90-17, August 1990.
7. *The TTL Logic Data Book*, Texas Instruments Inc., 1988.
8. *Data Sheet, STEL-1172B 50 MHz, 32-bit CMOS NCO*, Integrated Circuits Frequency Synthesis Products.

Appendix A: Ground terminal processor interface board user's guide

A1. Installation

The GT processor i/f board, which drives an HSC for the uplink synchronization experiments, is controlled by the GT processor. For the GT processor connection, the GT processor i/f board is but one of the slave boards which employs DSPLINK in the uplink synchronization experiments. As a result, a DSP backplane cage was built to accommodate multiple boards in the subsystem using DSPLINK. The 96-pin DSP backplane and the interface configuration are described in Sections 2.2.4.1 and 3.2.6.1 respectively. The GT processor i/f board is installed by inserting the end of the board with the J3 connector into one of the slots of the 96-pin backplane chassis. The DSP backplane is subsequently connected to the GT processor (DSP board) by a 50-pin ribbon cable and DSPLINK extender card. The extender card, which is inserted into one of the other slots, maps the fifty lines of DSPLINK to the 96-line backplane. The 50-pin ribbon cable should be no longer than 1 m.

On the other end of the GT processor i/f board, there are two connections to the HSC: the HSC command interface connection (J1); and the HSC data interface connection (J2). A 26-pin ribbon cable and a 10-pin ribbon cable are used respectively to connect the GT processor i/f board to the corresponding connectors on the back of the HSC box.

A2. Configuration and Reset

A2.1 GT processor i/f board jumper settings

As shown in Fig. 3.1., the GT processor i/f board contains three jumpers, labeled HD1, HD2, and HD3, which must be configured. HD1 is used to select the base address of the GT processor i/f board. HD2 is used to select whether the IOE* signal from the DSPLINK interface goes through a buffer before being used in the address decoder circuitry. HD3 is used to select one of two available data clocks generated by the divider circuit. Table A1 shows the jumper options. The selections indicated by the bar across two terminals and are described in **bold** are those used for the uplink synchronization trials.

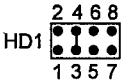
	Base address selection 1-2: Base address = 0 **3-4: Base address = 4** 5-6: Base address = 8 7-8: Base address = 12
	/IOE buffering **1-2: /IOE buffered** 3-4: /IOE not buffered
	Data clock selection 1-2: Data clock = 4800 b/s **3-4: Data clock = 2400 b/s**

Table A1. GT processor i/f board jumper settings

A2.2 GT processor i/f board potentiometer settings

The GT processor i/f board also contains two potentiometers which are used to set the pulse width of the multivibrators generating the STROBE signal for the HSC (see Fig. 3.1 and Fig. 3.2). These are labeled, POT1 and POT2. POT1 sets the pulse width of the STROBE ENABLE signal, while POT2 sets the pulse width of the STROBE pulse. They are currently set at 8.6 kΩ and 14.6 kΩ, yielding pulse widths of 59.6 ns and 100 ns respectively.

A2.3 Software reset of the GT processor i/f board

Although a RESET signal is generated on DSPLINK upon power up of the DSP board, a software reset is also made available for the GT processor i/f board. The active-low software reset is generated by writing a "0" to bit D2 (default set to "1") of the COMMAND register and then setting it back to the default. The result is to reset the NCO, the strobe generation circuit, the clock divider circuit, the data underflow circuit, and the interrupt generation circuit. It is recommended that a software reset be issued to the GT processor i/f board prior to operation. A sample code written for the Spectrum Signal Processing Inc TMS320C30 DSP board is provided below:

```
; Sample code for software reset written for the C30 DSP board

**** GT processor i/f board address ****

COMMAND         .set            800004h         ;Write only

**** GT processor i/f board commands ****

SW_RES_GT       .set            1               ;command register = 0000 0000 0000 0001
Dfault_CMD      .set            5               ;command register = 0000 0000 0000 0101

**** Data page pointers ****

DSPLNKPG        .set            80h

; Sample code begins here

                LDI             DSPLNKPG,DP     ;change data page pointer
                LDI             SW_RES_GT,R0    ;load s/w reset command into register
                STI             R0,@COMMAND     ;store register to COMMAND port

                LDI             Dfault_CMD,R0   ;load default command into register
                STI             R0,@COMMAND     ;store register to COMMAND port

; End of sample code
```

A3. Examples

The following sections provide examples for each of the functions of the GT processor i/f board.

A3.1 Clock generation

The NCO uses a phase increment, provided by the user, to produce the desired output frequency. The phase increment is entered by writing four consecutive 8-bit words to the NCO through the NCO and COMMAND ports of the GT processor i/f board. Each 8-bit word is identified by the ADDR0 and ADDR1 lines and must be specified when the user transfers the data value. When the entire phase increment has been transferred, a LDSTB command is issued to the NCO through the COMMAND register to start the NCO.

The paragraphs below demonstrate the steps followed to set the NCO to a particular frequency, f_o. It is assumed that the NCO has already been reset properly. The clock frequency of the NCO, f_c, is 40 MHz on the GT processor i/f board. Let f_o = 200 kHz for this example.

Step 1: Using the equation given in Section 3.2.2 (equation (1)), compute $\Delta\phi$.

$$\Delta\phi = \frac{2^{32}}{f_c} \times f_o = \frac{2^{32}}{40\times10^6} \times 200\times10^3 = 21474836 = 147AE14\,\text{h}$$

Step 2: Transfer the $\Delta\phi$ value to the NCO phase register in 8-bit blocks by transferring the 8-bits to the NCO port and then issuing an active-low WRN pulse on the COMMAND register (bit D0) of the GT processor i/f board. Each 8-bit block is identified by two "address" bits, ADDR0 and

ADDR1 which are transferred with each 8-bit word in bits D8 and D9 of the DSPLINK data lines and are defined as follows [8]:

ADDR1	ADDR0	$\Delta\phi$
0	0	D24-D31(MSB)
0	1	D16-D23
1	0	D8-D15
1	1	D0(LSB)-D7

Table A2. Phase register address assignments

Thus, for this example, to transfer $\Delta\phi$, the following four data words are transferred to the NCO using the NCO port, each followed by a WRN pulse on the COMMAND register,

D15... D0
0001 h
0147 h
02AE h
0314 h

Table A3. Four consecutive data words to be sent to the NCO

Step 3: Transmit an active-high LDSTB pulse using the COMMAND register of the GT processor i/f board (bit D1) to transfer the 32-bit $\Delta\phi$ register to the phase accumulator of the NCO. The LDSTB pulse starts the NCO and the required digitized sine wave with frequency f_o is produced after the 34 clock cycle pipeline delay specified by the NCO data sheet [8].

The following provides the sample code for the steps outlined above. Again, the code was written for the TMS320C30 DSP board.

```
; Sample code for NCO operation written for the C30 DSP board

**** Data page pointers ****

BSSPG           .set            80h
DATAPG          .set            0h
DSPLNKPG        .set            80h

**** GT processor i/f board addresses

; Base address of 4 assumed

COMMAND         .set            800004h         ;Write only
NCO_CMD         .set            800005h         ;Write only

**** NCO constants and commands ****

                .data
NCO_INIT        .float          1.92e5
NCO_const       .float          1.0737418e2
```

```
D0_D7mask       .word           0FFh
D8_D15mask      .word           0FF00h
D16_D23mask     .word           0FF0000h
D24_D31mask     .word           0FF000000h
D0_D7addr       .word           3000000h
D8_D15addr      .word           2000000h
D16_D23addr     .word           1000000h
D24_D31addr     .word           0h
Addr_Phase      .word           Phase

NCO_WRN_LO      .set            4h
NCO_STRB_HI     .set            7h
Dfault_CMD      .set            5h

**** Reserve memory ****

                .globl          Phase
                .bss            Phase,4
                .globl          phs_rnded
                .bss            phs_rnded,1
                .globl          index
                .bss            index,1

; Sample code begins here
; It is assumed that phs_rnded has been initialized with Δφ = 147AE14h

                LDI             BSSPG,DP
                LDI             @phs_rnded,R0
                LDI             DATAPG,DP
                AND             @D24_D31mask,R0         ;Mask out D24-D31 of phase register
                LSH             -8,R0                   ;Format for DSPLINK data
                ADDI            @D24_D31addr,R0         ;Append ADDR0 and ADDR1 values
                LDI             BSSPG,DP
                STI             R0,@Phase               ;Store in array


                LDI             BSSPG,DP
                LDI             @phs_rnded,R0
                LDI             DATAPG,DP
                AND             @D16_D23mask,R0         ;Mask out D16-D31 of phase register
                ADDI            @D16_D23addr,R0         ;Append ADDR0 and ADDR1 values
                LDI             BSSPG,DP
                STI             R0,@Phase+1             ;Store in array

                LDI             BSSPG,DP
                LDI             @phs_rnded,R0
                LDI             DATAPG,DP
                AND             @D8_D151mask,R0         ;Mask out D8-D15 of phase register
                LSH             8,R0                    ;Format for DSPLINK data
                ADDI            @D8_D15addr,R0          ;Append ADDR0 and ADDR1 values
                LDI             BSSPG,DP
                STI             R0,@Phase+2             ;Store in array


                LDI             BSSPG,DP
                LDI             @phs_rnded,R0
                LDI             DATAPG,DP
                AND             @D0_D7mask,R0           ;Mask out D0-D7 of phase register
                LSH             16,R0                   ;Format for DSPLINK data
                ADDI            @D0_D7addr,R0           ;Append ADDR0 and ADDR1 values
                LDI             BSSPG,DP
                STI             R0,@Phase+3             ;Store in array

                LDI             BSSPG,DP
                LDI             0,R0
                LDI             R0,IR0
                STI             R0,@index

                LDI             DATAPG,DP
                LDI             @Addr_Phase,AR0
Phase_trans:                                            ;Loop to transfer 4 segments of phase reg
                LDI             *+AR0(IR0),R1
                LDI             DSPLNKPG,DP
                STI             R1,@NCO_CMD             ;Transfer 8-bits of phase reg to i/f bd
```

```
                LDI             NCO_WRN_LO,R2
                LSH             16,R2
                STI             R2,@COMMAND             ;Send WRN pulse, load 8-bit segment into NCO
                LDI             Dfault_CMD,R2
                LSH             16,R2
                STI             R2,@COMMAND
                LDI             BSSPG,DP
                LDI             @index,R0
                ADDI            1,R0
                LDI             R0,IR0
                STI             R0,@index
                CMPI            4,R0
                BLT             Phase_trans

                LDI             DSPLNKPG,DP
                LDI             NCO_STRB_HI,R2
                LSH             16,R2
                STI             R2,@COMMAND             ;Send LDSTRB pulse to start NCO
                LDI             Dfault_CMD,R2
                LSH             16,R2
                STI             R2,@COMMAND

; End of sample code
```

A3.2 Commanding the hopping synthesizer controller

Commands for the HSC consist of writing 16-bit words to the HSC I/O port of the GT processor i/f board. In order for the HSC to properly read and execute the command, the GT processor must wait for the NReady bit, which is generated by the HSC, to be "low". After a 16-bit command word is transferred to the GT processor i/f board by the DSP, a strobe signal is generated on the i/f board to signal to the HSC that a command is ready to be serviced. While the HSC is servicing the command, the NReady bit is set to "high" by the HSC. Further details on commanding the HSC are provided in [2].

As an example, to load the base frequency parameter, a 16-bit command (030C h) is transferred, followed by two 16-bit words representing the 32-bit base frequency value, with the lower sixteen bits transferred before the upper sixteen bits. For the HSC, the base frequency value is in units of 100 Hz. Thus, to transfer a base frequency parameter of f_{base} = 44.625 GHz /100 Hz = 1A993C10 h, the following steps are required:

Step 1: Read the status register of the GT processor i/f board to check for the NReady bit (bit D2 and labelled as NRDY on the schematic) to be "low".

Step 2: When the Nready bit is found to be "low", send the 16-bit command (030C h) to the HSC port of the GT processor i/f board.

Step 3: Wait until the NReady bit is "low" again, indicating the HSC has read and processed the previous command.

Step 4: Send the lower sixteen bits of the base frequency parameter, 3C10 h, to the HSC port of the GT processor i/f board.

Step 5: Wait until the NReady is "low" again, indicating the HSC has read and processed the previous value.

Step 6: Send the upper sixteen bits of the frequency parameter, 1A99 h, to the HSC port of the GT processor i/f board.

The sample code for commanding the HSC as described above follows:

```
; Sample code for commanding the HSC written for the C30 DSP board

**** Data page pointers ****

BSSPG           .set            80h
DATAPG          .set            0h
DSPLNKPG        .set            80h

**** GT processor i/f board addresses ****

; Base address of 4 assumed

STATUS          .set            800004h         ;Read only
HSC_PORT        .set            800006h         ;Write only

**** HSC commands and miscellaneous ****

NRDY_BIT        .set            4               ;Bit D2 on GT status register

STOP_HSC        .set            0               ;STOP command for HSC
LD_BASE         .set            030Ch           ;Load base frequency of HSC command

; For this example, the base frequency is defined using the following constants
; Base frequency = 44.625 GHz/100 Hz = 1A993C10h

BASE_L16        .set            3C10h           ;Lower 16 bits of base frequency
BASE_H16        .set            1A99h           ;Upper 16 bits of base frequency

; Sample code begins here

NRDY_LOOP1:
                LDI             DSPLNKPG,DP
                LDI             @STATUS,R0      ;Read status register
                LSH             -16,R0
                TSTB            NRDY_BIT,R0     ;Check NReady bit
                BNZ             NRDY_LOOP1      ;Keep waiting if bit is "high"

                LDI             LD_BASE,R0
                LSH             16,R0
                STI             R0,@HSC_PORT    ;Send LD_BASE command to GT processor i/f board

NRDY_LOOP2:
                LDI             @STATUS,R0      ;Read status register
                LSH             -16,R0
                TSTB            NRDY_BIT,R0     ;Check NReady bit
                BNZ             NRDY_LOOP2      ;Keep waiting if bit is "high"

                LDI             BASE_L16,R0
                LSH             16,R0
                STI             R0,@HSC_PORT    ;Transfer lower 16 bits of base frequency

NRDY_LOOP3:
                LDI             @STATUS,R0      ;Read status register
                LSH             -16,R0
                TSTB            NRDY_BIT,R0     ;Check NReady bit
                BNZ             NRDY_LOOP3      ;Keep waiting if bit is "high"

                LDI             BASE_H16,R0
                LSH             16,R0
                STI             R0,@HSC_PORT    ;Transfer upper 16 bits of base frequency
```

```
; End of sample code
```

A3.3 Sending transmit data to the hopping synthesizer controller

During data communications in the uplink synchronization experiments, transmit data is sent to the HSC every hop. As described in Section 3.2.5, the transmit data consists of an 8-bit word and is formatted as follows:

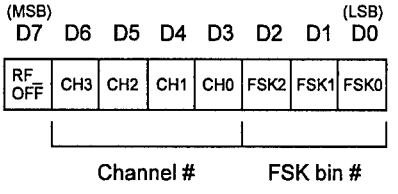


Fig. A1. Format of transmit data interface

A data underflow circuit implemented on the GT processor i/f board generates the signal for the underflow bit of the STATUS register to allow the GT processor to verify that data is continuously sent to the HSC. The GT processor must check the data underflow bit of the STATUS register (D3) (see Fig. 2.6) for a "low" which indicates that no underflow condition exists before sending the next FSK/frame data to the HSC.

An example to transfer the values, FSK bin 3, Channel 1, would follow the steps shown below:

Step 1: Read the STATUS register and test the data underflow bit for "low".

Step 2: If the data underflow bit is "low" transfer 000B h to the FSK/frame port of the GT processor i/f board. Otherwise, send an error message to the user indicating an underflow condition has occurred.

The sample code written for the TMS320C30 DSP board follows:

```
; Sample code for sending transmit data to the HSC written for the C30 DSP board

**** Data page pointers ****

BSSPG           .set            80h
DATAPG          .set            0h
DSPLNKPG        .set            80h

**** GT processor i/f board addresses ****

STATUS          .set            800004h         ;Read only
FSK_FRM         .set            800007h         ;Write only

**** HSC miscellaneous ****
```

```
UFLO_BIT        .set            8               ;Bit D3 on GT status register

; For this example, the transmit data to be transferred to HSC, 000B h, is stored
; as a constant

TX_DATA         .set            0Bh             ;RF_OFF=0, CHANNEL=1, FSK=3

; Sample code begins here

                LDI             DSPLNKPG,DP
                LDI             @STATUS,R0      ;Read status register
                LSH             -16,R0
                TSTB            UFLO_BIT,R0     ;Check underflow bit
                BZ              SEND_DATA

                CALL            UFLO_MSG        ;Call subroutine to send error message

SEND_DATA:
                LDI             TX_DATA,R0
                STI             R0,@FSK_FRM     ;Transfer data to i/f board

; End of sample code
```

A3.4 Interrupts

Once the interrupt registers have been initialized and interrupts have been enabled on the DSP board, an interrupt signal is generated by the i/f board to the GT processor on the rising edge of the hop clock. During an interrupt, the GT processor performs the necessary operations relating to the synchronization process. These operations are effected through an interrupt service routine. The GT processor must also clear the interrupt signal on the GT processor i/f board. The latter is performed by reading the INTRPT_PORT of the GT processor i/f board.

The sample code for clearing an interrupt generated by the GT processor i/f board is given below:

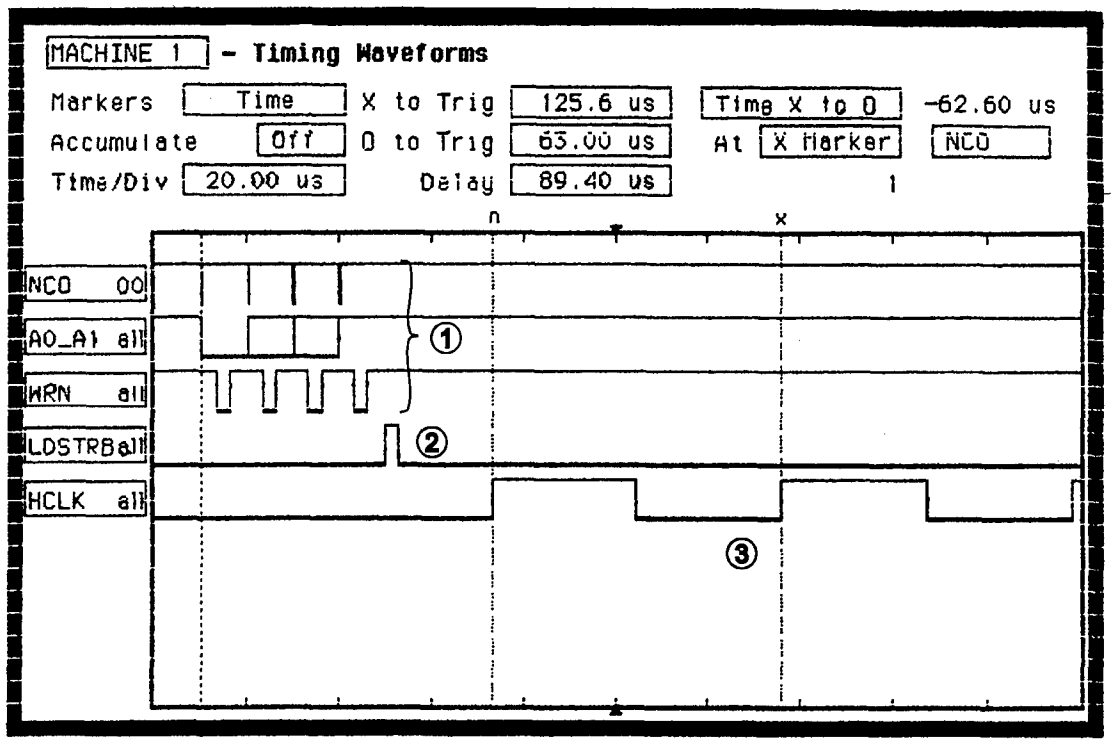
```
; Sample code for clearing interrupt written for C30 DSP board

**** Data page pointers ****

DSPLNKPG        .set            80h

**** GT processor i/f board addresses ****

; Base address of 4 assumed

INTRPT_PORT     .set            800004h

; Sample code begins here

                LDI             DSPLNKPG,DP
                LDI             INTRPT_PORT,R0          ;Read interrupt port to clear interrupt
                                                        ;Data from port is discarded

; End of sample code
```

Appendix B: Testing of the ground terminal processor interface board

B1. General

The GT processor i/f board was tested to ensure the proper operation of the four functions: hop clock and data clock signal generation (including interrupt pulse generation); HSC command interface; HSC data interface; and COMMAND and STATUS register operation . In this case, a Spectrum TMS320C30 DSP board is used to implement the test programs. An HP1650 Logic Analyser is used to verify the proper operation of the board. This appendix contains printouts of the logic analyser display for the first three operations of the i/f board. The COMMAND and STATUS register operations are verified by virtue of the proper operation of the first three functions.

B2. Clock generation



Legend:
NCO - Active-low NCO port select line of GT processor i/f board
A0_A1 - Addr0 and Addr1 lines of NCO
WRN - Active-low WRN pulse for NCO
LDSTRB - Active-high LDSTRB pulse for NCO
HCLK - Hop clock

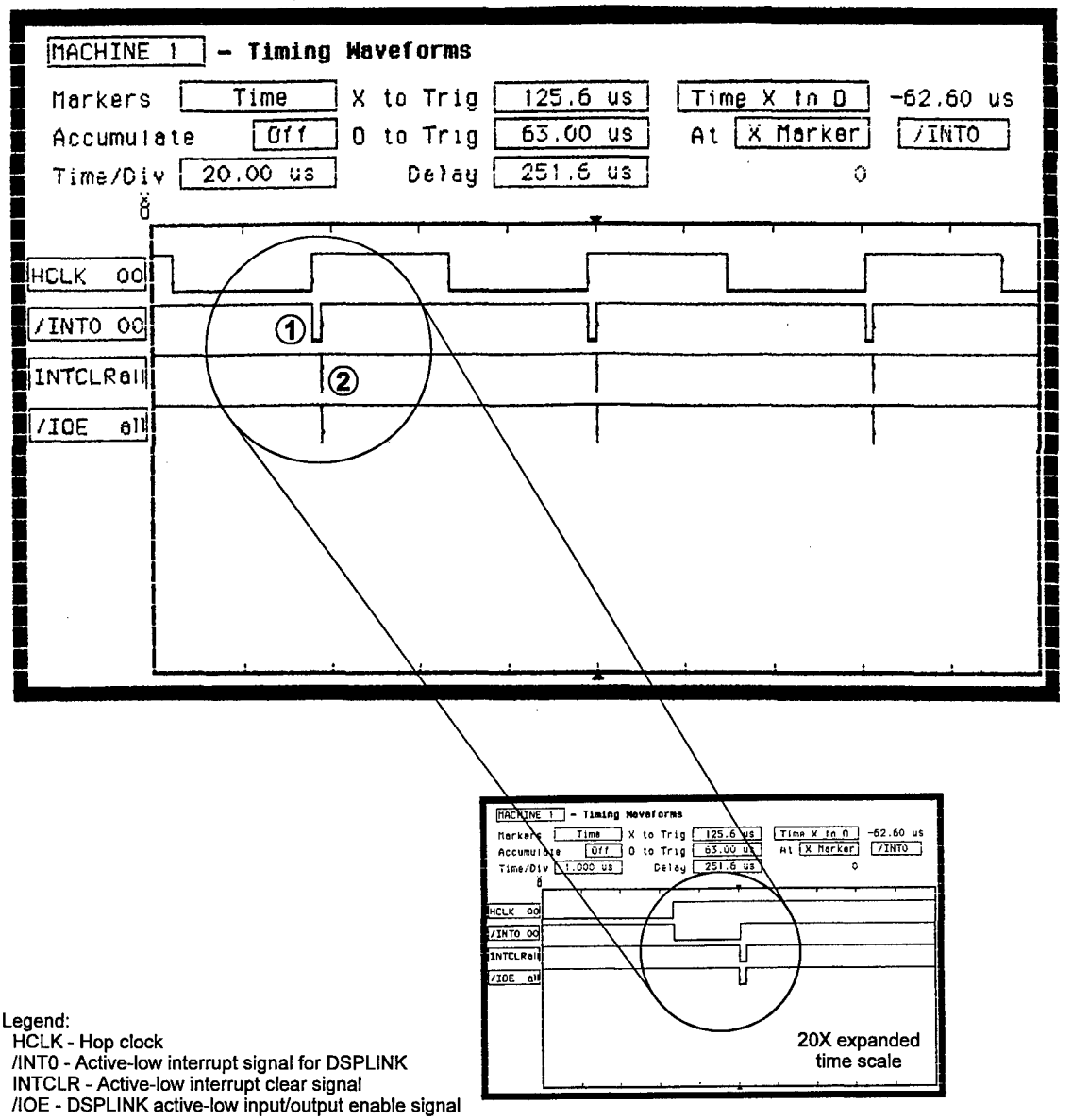
① Four consecutive NCO commands to load 32-bit phase increment value.
A0-A1 identify the 8-bit word of the 32-bit NCO phase increment.
WRN pulse to load the 8-bit word into the NCO phase register.

② Final LDSTRB pulse to transfer the entire 32-bit word to the NCO phase accumulator.

③ Hop clock generated from the NCO and subsequent divider circuit.
Hop clock period = 62.5 μs

Fig. B1 Verification of the clock generation circuit on the logic analyser

B3. Interrupt generation



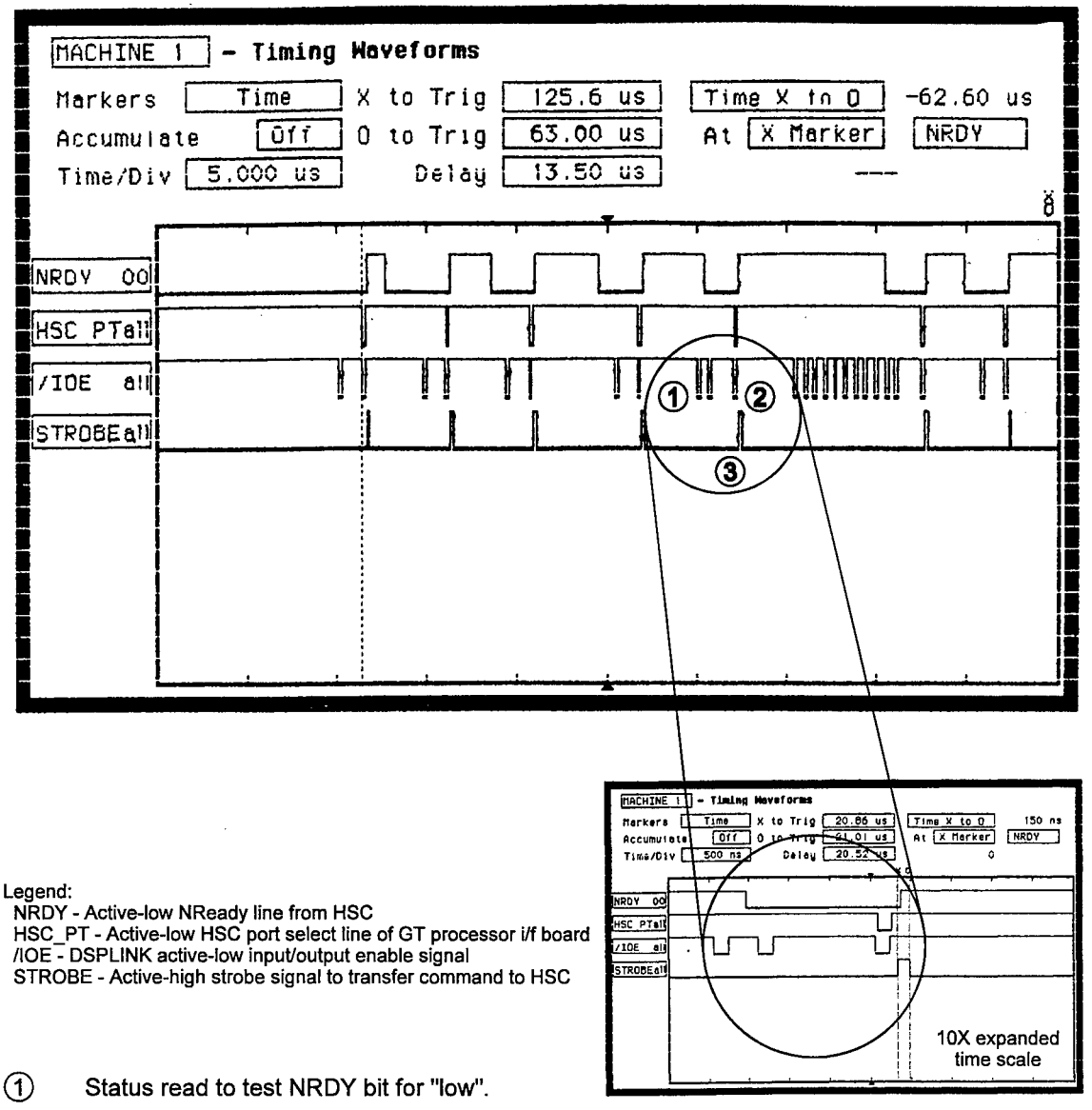
Legend:
HCLK - Hop clock
/INT0 - Active-low interrupt signal for DSPLINK
INTCLR - Active-low interrupt clear signal
/IOE - DSPLINK active-low input/output enable signal

① Interrupt signal generated on rising edge of hop clock.

② Interrupt is cleared by reading INTRPT_PORT (IOE access).

Fig. B2 Verification of interrupt circuit using the logic analyser

B4. HSC command and strobe generation



Legend:
NRDY - Active-low NReady line from HSC
HSC_PT - Active-low HSC port select line of GT processor i/f board
/IOE - DSPLINK active-low input/output enable signal
STROBE - Active-high strobe signal to transfer command to HSC

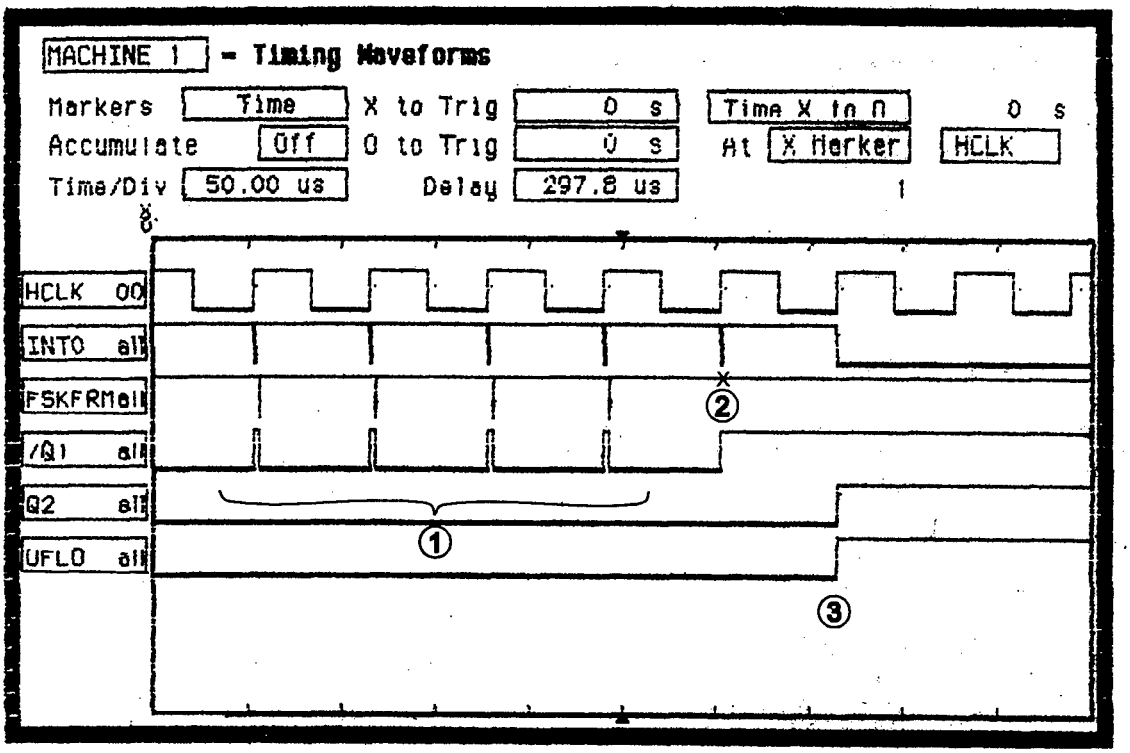
① Status read to test NRDY bit for "low".

② HSC command sent.

③ Strobe pulse generated (minimum pulse width = 50 ns).

Fig. B3 Verification of the hopping synthesizer controller command and strobe generation circuit using the logic analyser

B5. HSC transmit data interface and underflow circuit operation



Legend:
HCLK - Hop clock
INT0 - Active-low interrupt signal
FSK_FRM - Active-low FSK/FRAME port select line of GT processor i/f board
/Q1 - Inverted output of first flip flop of underflow detect circuit
Q2 - Output of second flip flop of underflow detect circuit
UFLO - Dat_UFLO signal

① FSK/Frame data issued on every hop after reading the STATUS register and testing the UFLO bit (latter not shown).

② No FSK/Frame data written to FSK/frame port for this hop.

③ Data underflow bit set on the following hop.

Fig. B4 Verification of the transmit data underflow circuit

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)
Defence Research Establishment Ottawa
Ottawa, Ontario
K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable)
UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)
Ground Terminal Processor Interface Board for Skynet Uplink Synchronization Trials (U)

4. AUTHORS (Last name, first name, middle initial)
Tom, C.

5. DATE OF PUBLICATION (month and year of publication of document)
November 1997

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)
61

6b. NO. OF REFS (total cited in document)
8

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.
DREO Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.
SST, Defence Research Establishment Ottawa
Ottawa, Ontario, K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)
MITI Thrust

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)
DREO REPORT 1321

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). however, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)
Unlimited Announcement

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

RA.W (24 Nov 93)

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

A ground terminal (GT) simulator subsystem is being developed at Defence Research Establishment Ottawa (DREO) as part of the in-house work examining the aspects of uplink synchronization for extremely-high-frequency (EHF) spread spectrum satellite communications (SATCOM). Requirements of the GT subsystem include the generation of hop clock and data clock signals, and the interface between the GT processor and a hopping synthesizer controller (HSC) for commanding the HSC and transmitting data. A GT processor interface (i/f) board was designed and fabricated at DREO to satisfy these requirements. This report describes the functions of the i/f board and specific requirements related to the uplink synchronization experiments and the interface to the HSC. The i/f board is a printed circuit board which is contained in a backplane chassis and is driven by the GT processor. The GT processor is realized by a Spectrum Signal Processing Inc. TMS320C30 digital signal processor board and communicates with the GT processor i/f board via the DSPLINK interface through the backplane. This report includes implementation details of the clock generation and interface circuitry and a user's guide for the proper configuration, installation and operation of the GT processor i/f board.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

ground terminal simulator interface board, uplink synchronization, clock generation